Université de Montréal

MC-Map, un nouvel outil d'intégration de motifs

par

Nicolas St-Onge

Département de biochimie, Université de Montréal

Faculté des Études Supérieures

Mémoire présenté à la Faculté des études supérieures

en vue de l'obtention du grade de Maîtrise

en bio-informatique

Décembre 2006

Université de Montréal
Faculté des études supérieures

Ce mémoire intitulé :

MC-Map, un nouvel outil d'intégration de motifs

présenté par :
Nicolas St-Onge

a été évalué par un jury composé des personnes suivantes :

Serguei Chteinberg, président-rapporteur
François Major, directeur de recherche
Pascal Chartrand, membre du jury

Résumé

Avec l'apparition de modèles de cristallographie de macromolécules d'ARN telles les sous-unités ribosomales, il est devenu plus complexe d'analyser la structure de l'ARN et de rechercher des motifs d'intérêts. Si certains outils de recherche de motifs se restreignent à la recherche de motifs spécifiques (tel les tiges boucles, les ARNt, etc.), d'autres offrent une flexibilité, tel l'outil *MC-Search* du laboratoire LBIT, permettant de rechercher un motif à partir d'un descripteur de motif.

Nous avons envisagé que l'étude de motifs structuraux nécessite une plateforme d'intégration. Les instances de motifs trouvées doivent être stockées pour pouvoir être référées et analysées ultérieurement. L'application Web *MC-Map* a été créé en ce sens, permettant l'intégration d'instances de motifs, en plus de cartographier les résultats. En effet, tout utilisateur peut visionner les résultats sous forme de représentations 2-D de brins d'ARN que nous appelons cartes d'ARN. Ces cartes permettent de localiser les instances de motifs et donc de mieux évaluer leur fonction. Le serveur Web se trouve à l'adresse http://www-lbit.iro.umontreal.ca/mcmap/.

Mots-clés : ARN, BASE DE DONNÉES, INTÉGRATION DE DONNÉES, MOTIFS, SERVEUR WEB, STRUCTURE

Abstract

With the apparition of RNA macromolecules crystallographic models such as the ribosomal sub-units, RNA analysis and structural motif search has become an arduous task. Thus, tools have been created for searching RNA structural motifs from these models. The application *MC-Search*, developed from our lab, enables the search of any user defined motif, given a description of this motif.

We believe that structural motif analysis requires an integration platform. Found motif instances need to be stored in such platform, so that they can be viewed and analysed later on by any user. The web application *MC-Map* has been implemented for mapping motif instances to PDB structures. Results are viewed in 2D graphical representation called RNA maps, revealing motif localization in RNA strands, as well as insights in the motif function. The web server can be accessed at http://www-lbit.iro.umontreal.ca/mcmap/.

Keywords: DATA INTEGRATION, DATABASE, MOTIFS, RNA, STRUCTURE, WEB SERVER

Table des matières

Liste des tableaux

Liste des figures

À ma blonde, qui m'a soutenu durant ma maîtrise et a fait preuve de patience...

Remerciements

J'aimerais tout d'abord remercier mon directeur de recherche, François Major, qui m'a aidé à réaliser ce projet et ce, malgré son horaire plus que débordé. Je tiens également à remercier Martin Larose pour son support informatique, notamment sur l'application *MC-Search*, ainsi qu'Emmanuelle Permal pour avoir fourni sa classification de structures PDB. Finalement une pensée pour tous les collègues du laboratoire qui ont créés une ambiance au laboratoire propice à l'entraide et la décontraction (Amine Halawana, Jean-Phillipe Doyon, Louis-Philippe Lavoie, Mahshid Shakiba, Marc Parisien, Maribel Hernandez, Mohamed Tikah, Philippe Thibault, Sébastien Cristen).

Chapitre 1 : Introduction à la structure de l'ARN

Depuis quelques décennies, un des objectifs de la recherche sur la structure de l'ARN est de prédire la fonction d'un brin d'ARN en fonction de sa séquence. En effet, la complexité de la prédiction du repliement de l'ARN est comparable à celle de la prédiction du repliement des protéines.

L'ARN est une molécule structurée de nucléotides où chaque nucléotide comporte une base azotée, un sucre et un phosphate. Les quatre bases azotées possibles des nucléotides sont l'adénine, l'uracile, la cytosine et la guanine. À noter que l'adénine et la guanine, par leur structure moléculaire, font partie du groupe des purines tandis que l'uracile et la cytosine font partie du groupe des pyrimidines. Le sucre contenu dans les nucléotides est toujours un ribose. Les nucléotides sont liés entre eux par un groupe phosphate ainsi qu'un ribose, ce qui permet à l'ARN d'être sous forme de longues chaînes. Les phosphates avec les riboses forment le squelette (backbone) de la molécule d'ARN.

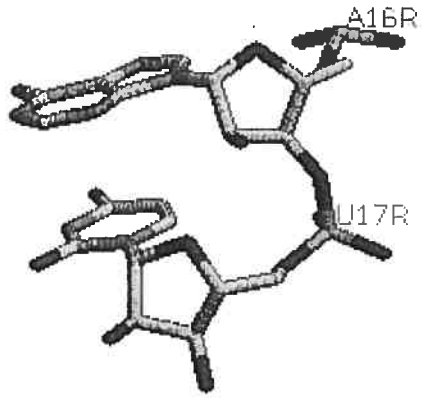


Figure 1: Chaîne d'ARN contenant une adénine suivie d'une uracile dans le sens 5'- 3'. Les deux nucléotides font parti de la chaîne R du fichier PDB 1AQ3, la molécule décrite étant un brin d'ARN en complexe avec une protéine de capside d'un bactériophage.

Les nucléotides interagissent entre eux pour former la structure secondaire et tertiaire de l'ARN. Ainsi, est-il possible de définir la structure de l'ARN en fonction de ces interactions nucléotidiques : c'est ce que nous appelons l'annotation de molécules d'ARN. Bien qu'il existe plusieurs types d'interaction, les principales interactions demeurent les liaisons covalentes d'adjacence, l'appariement de bases et les empilements de bases (stacking). Nous pourrions énumérer d'autres types d'interaction nucléotidique tel

l'interaction d'un groupement phosphate avec une uridine que l'on retrouve dans le motif U-turn des ARN de transfert. Nous nous attarderons par contre aux trois interactions principales énumérées.

La liaison covalente d'adjacence, plus souvent appelé l'adjacence, résulte de deux nucléotides qui se suivent consécutivement sur une même chaîne d'ARN. Ce sont des liaisons très fortes apportant une stabilité à la molécule. L'appariement de bases est l'attraction que deux bases azotées de deux nucléotides différents peuvent avoir un envers l'autre. L'appariement est traduit par la formation de ponts hydrogènes. Si chaque paire de bases possible (4 bases azotées x 4 bases azotées, donc 16 paires possibles en tout) peut avoir un certain degré d'appariement, toutes les paires de bases n'ont certes pas le même degré d'attraction. Également, une même paire de bases azotées peut avoir plusieurs conformations géométriques possibles tout dépendamment comment les bases azotées se retrouvent dans l'espace 3D, ce qui viendra influencer le degré d'appariement. La force de l'appariement est directement proportionnel au nombre de ponts hydrogènes formés. Ce nombre peut varier entre un et trois. Les appariements les plus fréquents et les plus connus sont les appariements de type canonique, soit les appariements cytosine-guanine (C-G) et les appariements adénosine-uracile (A-U). Ces appariements, impliquant une purine avec une pyrimidine, comportent respectivement trois et deux ponts hydrogènes, ce qui vient renforcir la stabilité de la molécule d'ARN.

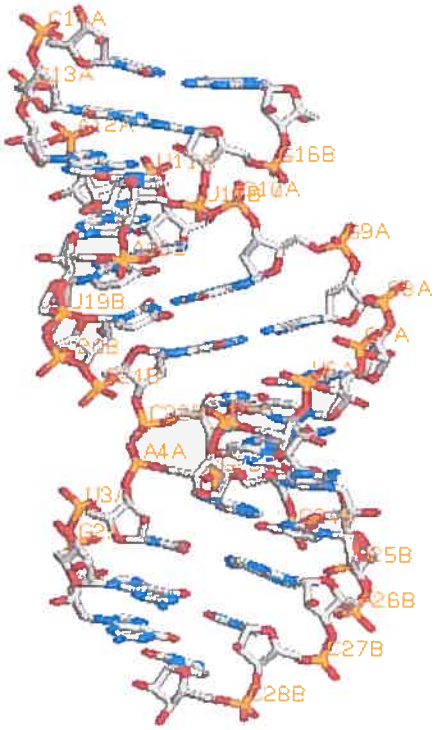


Figure 2: Double hélice d'ARN tel que nous pouvons l'observer dans le fichier PDB 433D, décrivant une molécule d'ARN double-brin. Cette double hélice est formée à partir de plusieurs d'appariements de base, mais également d'empilements de base. Les tiges dorées représentent les atomes de phosphate, les tiges rouges dénotent les atomes d'oxygène, les tiges bleues illustrent les atomes d'azote et finalement les tiges grises représentent les atomes de carbone.

Certaines structures d'ARN telle la large sous-unité ribosomale contiennent environ 50% de paires non-canoniques. Ces paires non-canoniques sont toutes les paires possibles de bases autres que les paires C-G et A-U. Des exemples de paires non-canoniques bien connues sont les paires dites Wobble (1), liant une guanine avec une uracile dans une conformation géométrique particulière et les paires dites Sheared (1), liant une guanine avec une adénine, également dans une conformation géométrique bien particulière. Ces deux paires comportent deux ponts hydrogènes, apportant donc une stabilité à la molécule. Enfin, toutes les autres paires possibles font partie des paires non-canoniques. La plupart de ces paires étant particulièrement rares, elles suscitent moins d'intérêt.

Les appariements peuvent être décrits plus formellement en nommant les faces mises en jeu de chaque nucléotide de l'interaction. Ces faces sont appelées Watson-Crick, Hoogsteen et Sucre (2). Les interactions Watson-Crick sont les plus fréquentes. Les deux prochaines figures illustrent les faces en question.

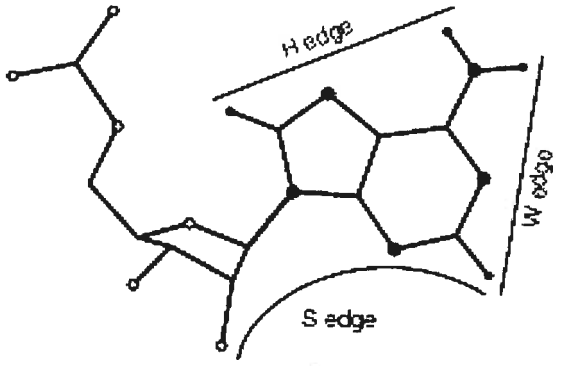


Figure 3: Chacune des trois faces d'une purine pouvant être impliquées dans un appariement. On retrouve en haut de la figure la face Hoogsteen dénotée par un H, sur le côté droit la face Watson-Crick dénotée par un W, et finalement la face Sucre au bas, dénotée par un H. Cette image a été rognée à partir de l'image source suivante : http://www-lbit.iro.umontreal.ca/mcannotate/Fig3.gif

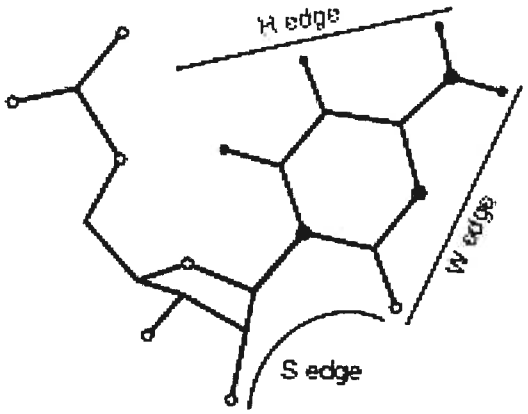


Figure 4: Chacune des trois faces d'une pyrimidine pouvant être impliquées dans un appariement. On retrouve en haut de la figure la face Hoogsteen dénotée par un H, sur le côté droit la face Watson-Crick dénotée par un W, et finalement la face Sucre au bas, dénotée par un H. Cette image a été rognée à partir de l'image source suivante : http://www-lbit.iro.umontreal.ca/mcannotate/Fig3.gif

Les empilements ont lieu le plus souvent entre des bases azotées adjacentes, dans des hélices double-brin. Ces empilements permettent d'augmenter la stabilité des molécules d'ARN et favorisent la forme hélicoïdale de l'ARN. La stabilisation des empilements implique des forces de dispersion de London (3) ainsi que des interactions entre charges partielles à l'intérieur des cycles aromatiques (4).

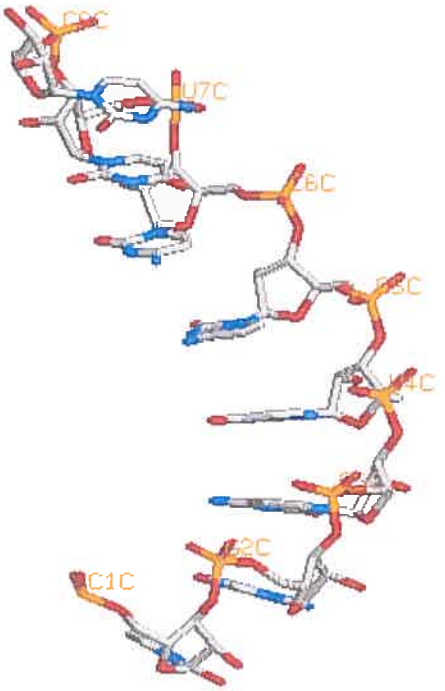


Figure 5: Empilements de bases azotées que l'on peut observer dans une hélice double-brin. À noter qu'un des deux brins de l'hélice a été masqué pour que les empilements soient plus facilement observables. Le brin d'ARN ici illustré correspond aux nucléotides 1 à 8 de la chaîne C du fichier PDB 1JBR, décrivant une structure cristallisée d'un complexe inhibiteur de restrictocine impliqué dans la reconnaissance d'ARN. Les tiges dorées représentent les atomes de phosphate, les tiges rouges dénotent les atomes d'oxygène, les tiges bleues illustrent les atomes d'azote et finalement les tiges grises représentent les atomes de carbone.

Une manière d'étudier la structure de l'ARN est par l'identification et l'étude de motifs. Les motifs sont par définition des séquences génomiques récurrentes. Le fait que les motifs soient récurrents indiquent le plus souvent une récurrence au niveau de la structure, et probablement une récurrence de fonction, d'où leur intérêt. Dans le contexte de la recherche sur la structure de l'ARN, nous nous intéressons davantage aux motifs

structuraux, c'est-à-dire des motifs dont la séquence similaire indique une structure similaire. En effet, notre hypothèse de départ est que la séquence de l'ARN détermine la structure de l'ARN qui elle-même détermine la fonction.

Les deux prochaines figures illustrent des exemples de petits motifs structuraux populaires, soit les tetraloop GNRA et UNCG. Un tetraloop peut être défini comme une boucle contenant quatre nucléotides non appariés par des appariements Watson-Crick et flanqué par un appariement Watson-Crick. Les lettres GNRA et UNCG réfèrent au code IUPAC où G représente une guanine, N représente un nucléotide quelconque, R une purine, A une adénine, U une uracile et C une cytosine.

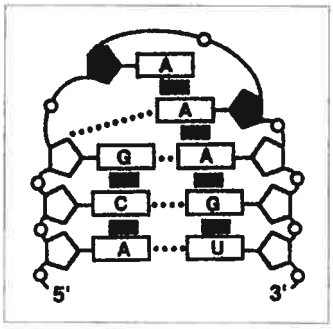


Figure 6 : Tetraloop de type GNRA ou plus exactement GAAA. Les pentagones illustrent les riboses, les petits cercles dénotent les groupes phosphate et les rectangles contenant une lettre illustrent les différentes bases. On retrouve entre les différentes bases des boîtes noires représentant les empilements ainsi que des pointillés représentant des appariements. La boucle du GNRA se trouve vers le haut de la figure, débutant avec une guanine appariée avec une adénosine par un appariement de type Hoogsteen-Sugar (2 pointillés). Le motif comporte souvent un double empilement dans la boucle, entre les trois adénosines du haut. Image provenant de *J. Nowakowski & I. Tinoco, Jr, RNA structure in solution, p 573*

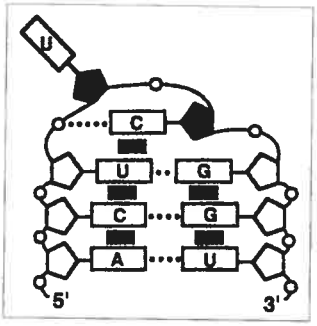


Figure 7: Tetraloop de type UNCG ou plus exactement UUCG. La boucle se retrouve également vers le haut de la figure. Ce motif comporte souvent un appariement non Watson-Crick entre les bases U et G (2 pointillés) ainsi qu'un empilement inversé entre U et C qui sont deux bases non adjacentes. Image provenant de *J. Nowakowski & I. Tinoco, Jr, RNA structure in solution, p 573*

Afin d'étudier les motifs structuraux, il est impératif d'avoir une source de données ainsi que des outils appropriés pour permettre l'analyse. La recherche sur la structure de l'ARN s'appuie considérablement sur la Protein Data Bank (PDB) (5), une base de fichiers nommés également par le même nom. Les fichiers PDB sont des fichiers texte contenant les coordonnées atomiques de molécules, voire macromolécules. Ces molécules sont le plus souvent des protéines, mais on retrouve également des molécules d'ARN ainsi que des complexes protéines-ARN. Les coordonnées atomiques de ces fichiers PDB sont le plus souvent fournies à partir de modèles de cristallographie aux rayons X ou de RMN.

```
ATOM    500  C1*   G B  24       4.869   4.770  27.082  1.00  9.72           C
ATOM    501  N9    G B  24       4.610   6.177  26.797  1.00 13.14           N
ATOM    502  C8    G B  24       3.389   6.817  26.791  1.00 15.37           C
ATOM    503  N7    G B  24       3.481   8.101  26.552  1.00  1.81           N
ATOM    504  C5    G B  24       4.845   8.319  26.374  1.00 14.33           C
ATOM    505  C6    G B  24       5.577   9.533  26.111  1.00  9.11           C
```

Figure 8: Quelques lignes de code du fichier PDB 433D décrivant les atomes 500 à 505 d'une structure de double hélice d'ARN. Le mot-clé ATOM, en début de rangée, vient nous renseigner que la ligne nous donne de l'information sur un atome particulier de la structure. Le nombre dans la deuxième colonne indique son numéro d'identification tandis que l'expression de la troisième colonne indique l'atome qu'il représente. Sur les 4ème, 5ème et 6ème colonnes se trouvent respectivement la nature du résidu a lequel appartient l'atome, le nom de la chaîne et le numéro identificateur du résidu au sein de la chaîne en question. Nous retrouvons par après sur les cinq prochaines colonnes les coordonnées X, Y, Z de l'atome ainsi que son occupation et facteur de température respectivement. Finalement, nous retrouvons en dernière colonne le symbole de l'élément. Le format des fichiers PDB est décrit plus en détails à l'url http://www.rcsb.org/pdb/file_formats/pdb/

Bien que les fichiers PDB soient des fichiers textes, il est possible, grâce à des logiciels de visualisation 3D, de pouvoir observer les molécules décrites dans ces fichiers dans une perspective 3D. Des exemples bien connus de ces logiciels sont Rasmol (http://www.umass.edu/microbio/rasmol/) et Pymol (http://pymol.sourceforge.net/). Ces logiciels peuvent appliquer des opérations de rotation, déplacement et d'agrandissement à la molécule étudiée, au gré de l'usager. On peut également colorier certaines parties de la molécule, effacer du champ d'observation des nucléotides sélectionnés et même détecter tous les nucléotides qui se retrouvent dans un rayon de distance donné à partir d'un nucléotide de référence. Par contre, ces logiciels ne permettent pas d'étudier les interactions nucléotidiques autre que par observation à l'oeil nu. La Figure 2 démontre un exemple d'une molécule d'ARN visualisée à partir du logiciel Rasmol.

Avec l'apparition de structures de cristallographie à rayons X de haute résolution de la large sous-unité ribosomale (7-9), il devient de plus en plus pertinent de recourir à des outils d'analyse d'ARN plus poussés que des simples logiciels de visualisation tel Rasmol. Au Laboratoire de Biologie Informatique et Théorique (LBIT), des outils ont été développés pour mieux analyser des structures d'ARN décrites dans les fichiers PDB. Un de ces outils, *MC-Annotate* (10), permet l'annotation de molécules d'ARN. Un autre outil dont nous référerons beaucoup plus souvent est *MC-Search* (11) et est utilisé pour rechercher des motifs structuraux dans les fichiers PDB. C'est en partie à cet outil que le prochain chapitre sera consacré.

Chapitre 2 : Recherche de motifs avec *MC-Search*

Nous souhaiterions rechercher un motif structural d'ARN bien précis dans une molécule contenant un ou plusieurs brins d'ARN décrit par un fichier PDB. La question que l'on peut se poser en un premier temps est la suivante : comment peut-on définir un motif structural en termes de mots ou expressions? Une solution serait de définir la structure primaire et secondaire de ce motif. La structure primaire pourrait suivre le code IUPAC tandis que la structure secondaire serait une énumération d'interactions nucléotidiques. Prenons par exemple le motif structural de la tetraloop de type GNRA que nous souhaitons rechercher et dont une instance est illustrée à la Figure 9 :

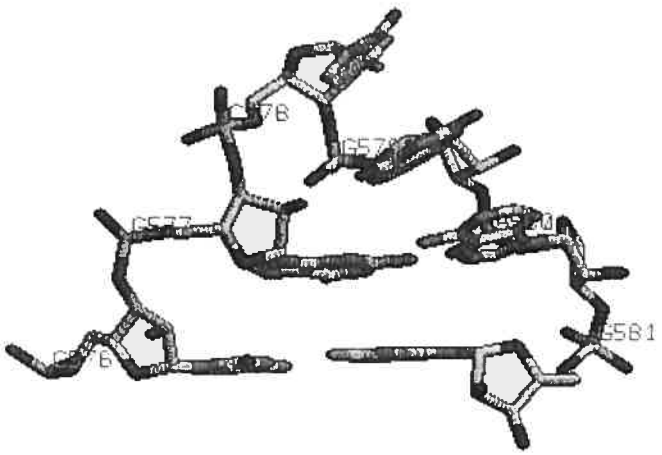


Figure 9: Instance d'un motif tetraloop, provenant d'une structure de la large sous-unité de l'organisme Haloarcula Marismortui, plus exactement de la chaîne 0, entre les résidus 576 et 581 inclusivement, de la structure PDB 1JJ2. Nous pouvons observer que les résidus à chaque extrême sont appariés ensemble, que la séquence interne est GCGA, ce qui fait de cette instance une instance de tetraloop GNRA. Finalement, les résidus G577 et A580 sont liés par un appariement Hoogsteen-Sugar.

Un motif tetraloop de type GNRA est défini ici comme une tige boucle flanquée par une paire canonique, où la séquence interne débute par une guanosine, suivie par n'importe quel nucléotide, d'une purine et finalement d'une adénosine, dans l'ordre 5'- 3'. Un appariement de type Hoogsteen-Sugar lie la première guanosine à la dernière adénosine de la séquence interne GNRA. Le motif est illustré par un croquis à la Figure 10:

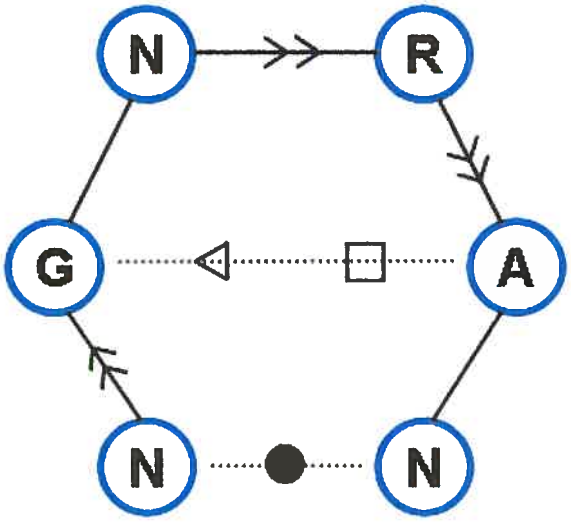


Figure 10: Illustration 2D d'un motif tetraloop GNRA. Chaque cercle bleu représente un nucléotide, chaque arête un lien d'adjacence tandis qu'un lien pointillé en noir démontre un appariement. Les lettres GNRA réfèrent au code UIPAC où N représente n'importe quel nucléotide, G représente une guanosine, R une purine et finalement A une adénosine. Les deux nucléotides du bas sont liés par un appariement de type Watson-Crick, illustré par un cercle rempli en noir. Les deux nucléotides du centre sont liés par un appariement de type Hoogsteen-Sugar représentés par un carré et un triangle vide. Les flèches sur les arêtes indiquent la présence d'empilement.

Ce motif pourrait être décrit de manière non formelle avec les spécifications suivantes:

Séquence :	NGNRAN
Structure secondaire :	• Le premier nucléotide de la séquence s'apparie avec le dernier selon un appariement Watson-Crick • La première guanine s'apparie avec la dernière adénine, dans la séquence interne GNRA • Trois empilements s'y retrouvent, soit entre le 1er et 2ème nucléotide, les 3ème et 4ème nucléotides, puis les 4ème et 5ème nucléotide de la séquence complète.

Table 1: Énumération des propriétés de structure primaire et secondaire du motif tetraloop GNRA

Une fois que nous avons formulé notre description de motif, nous savons donc ce que nous voulons rechercher exactement. Ceci dit, comment parvenir à trouver un tel motif dans des fichiers de structure PDB? Une solution serait d'observer cette molécule à l'oeil nu. Mais une telle solution comporte plusieurs inconvénients. Premièrement, il se pourrait que l'observateur omette certaines instances de motif, l'erreur étant humaine. Également, il pourrait être subjectif de déterminer qu'un arrangement de nucléotides respectent la description de motif décrite : par exemple, y a-t-il lieu de croire qu'il se trouve un appariement de bases ou même un empilement de bases entre tel et tel autre nucléotide dans notre boucle GNRA. Finalement, pour des fichiers de structure PDB décrivant des macromolécules tel les sous-unité ribosomales, il deviendrait extrêmement complexe de rechercher des motifs structuraux quelconques.

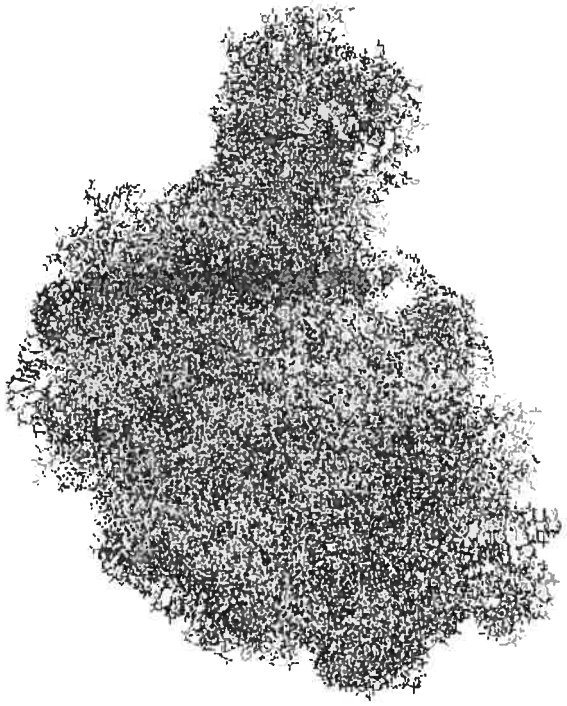


Figure 11: Illustration de la large sous-unité ribosomale de l'organisme Haloarcula marismortui décrite par le fichier de structure PDB 1JJ2, tel que visionné par le logiciel Rasmol. Trouver des instances de motif quelconque reviendrait à chercher une aiguille dans une botte de foin. L'utilisation de logiciels de recherche de motif s'avère pertinente.

Pour les raisons énumérées, il devient donc particulièrement utile de recourir à un outil informatique permettant la recherche des motifs structuraux dans des fichiers de structure PDB. Un tel outil a été conçu au LBIT : *MC-Search*.

Les fichiers de description de motif *MC-Search*

MC-Search utilise les concepts discutés précédemment. Afin de lancer une recherche de motifs, *MC-Search* requiert que l'usager décrive dans un fichier texte la description du motif d'intérêt. Également, l'usager doit spécifier quels seront les fichiers de structures PDB à traiter. Afin que *MC-Search* puisse interpréter la description de motif donnée, celle-ci doit suivre une syntaxe et nomenclature bien spécifique. Ainsi, la description du motif tetraloop GNRA, énoncée de manière non formelle à la Table 1, s'écrirait selon la syntaxe *MC-Search* de la manière suivante :

```
// Description du motif tetraloop GNRA
sequence (
    RNA A1 NGNRAN
)

relation (
    A1 A6 {W/W}
    A2 A5 {S/H}
    A1 A2 {stack}
    A3 A4 {stack}
    A4 A5 {stack}
)
```

Table 2: Descripteur *MC-Search* du motif tetraloop GNRA décrit antérieurement

Nous pouvons observer de ces lignes de code que le vocabulaire *MC-Search* contient des mots-clés en anglais, qui sont reliés à la structure. Examinons ce code en détails. La première ligne dénote un commentaire. Toutes les lignes débutant par une double barre oblique sont des lignes de commentaires et conséquemment, sont ignorées par l'interpréteur de *MC-Search*. La deuxième ligne vient débuter un premier bloc

d'instructions. Il y a en tout deux blocs d'instructions dans ce code. En effet, chaque bloc d'instruction débute par un mot-clé (dans ce cas-ci les mots 'sequence' et 'relation'), suivi d'une parenthèse ouvrante, d'une série de lignes d'instruction, et se termine par une parenthèse fermante. Notre premier bloc d'instructions, qui débute par le mot-clé 'sequence', se rapporte à la structure primaire de notre motif d'intérêt. Comme les 'tetraloop' ne comportent qu'un brin d'ARN, nous ne retrouvons qu'une seule ligne d'instruction à l'intérieur du bloc, soit :

```
RNA A1 NGNRAN
```

Sur cette ligne, trois paramètres viennent définir notre brin d'ARN. Premièrement, le premier paramètre vient donner l'information sur la nature de ce brin, dans ce cas-ci de l'ARN. Le deuxième paramètre vient nommer le brin ainsi que le premier nucléotide qui s'y retrouve. En effet, nous savons par ce paramètre que notre brin d'ARN est nommé A et que le premier nucléotide qui s'y retrouve dans l'ordre de séquence 5' – 3' est A1. Conséquemment, le deuxième nucléotide de notre brin d'ARN est nommé A2, notre troisième nucléotide est nommé A3 et ainsi de suite jusqu'au nucléotide A6, puisque nous avons en tout six nucléotides. De cette façon, nous pourrons référer à chacun des nucléotides pour définir des interactions nucléotidiques.

Étudions maintenant le dernier bloc d'instructions qui débute par le mot-clé 'relation' et se rapporte à la structure secondaire. Notons que dans le jargon *MC-Search,* le mot relation est utilisé comme synonyme à interaction nucléotidique. Nous avons en tout cinq relations définies dans ce bloc, soit :

```
A1 A6 {W/W}
A2 A5 {S/H}
A1 A2 {stack}
A3 A4 {stack}
A4 A5 {stack}
```

La première ligne informe l'interpréteur *MC-Search* que nous désirons un appariement de type Watson-Crick entre le premier et le dernier nucléotide de notre brin

d'ARN. Pareillement, la deuxième ligne vient informer que le deuxième et cinquième nucléotide sont appariés par un lien Sugar-Hoogsteen. *MC-Search* permet d'ajouter de l'information sur la nature de cet appariement. Par exemple, si nous voulons que cet appariement soit canonique ou non, ou même définir les faces impliquées dans l'appariement. Finalement, les trois dernières lignes viennent informer qu'un empilement se retrouve respectivement entre les nucléotides A1 et A2, A3 et A4 ainsi que A4 et A5.

Afin de connaître plus en détails le vocabulaire utilisé par *MC-Search* et sa syntaxe, une page Web de type Wiki a été conçu à cet effet sur le site du LBIT à l'adresse http://www-lbit.iro.umontreal.ca/wiki/index.php/MC-Search

Fonctionnement de *MC-Search*

Une fois que nous avons rédigé notre descripteur de motif, *MC-Search* valide sa syntaxe avant de démarrer la recherche du motif. Si la validation échoue, un message s'affiche et donne de l'information sur la nature des erreurs trouvées. Dans le cas opposé, *MC-Search* enclenche son processus de recherche. C'est alors que l'outil *MC-Annotate* (10) entre en jeu en annotant les fichiers de structure PDB sélectionnés, produisant ainsi les éléments d'un graphe structurel. *MC-Annotate* transforme également le descripteur de motif en graphe structurel, ce graphe étant le graphe cible qui sera recherché selon l'algorithme d'isomorphisme d'Ullmann (12) par *MC-Search*. Les fragments ou arrangements de nucléotides retenus par *MC-Search* sont ceux répondants aux exigences du graphe cible.

Le graphe structurel produit pour chacun des fichiers PDB ainsi que pour le descripteur contient les informations géométriques sur les conformations nucléotidiques et les interactions inter bases, les coordonnées atomiques ainsi que les angles de torsion (10). Des symboles sont calculés et assignés à chacune des conformations nucléotidiques et interactions inter bases du graphe structurel. L'utilisation de symboles pour représenter des

structures d'ARN plutôt que de coordonnées atomiques et d'angles de torsion simplifie grandement la comparaison de structures d'ARN ainsi que la reconnaissance du motif d'intérêt recherché dans les fichiers de structure PDB.

Une fois que *MC-Search* a terminé sa recherche de motifs, il créé des fichiers PDB de sortie de toutes les instances de motif trouvées. La taille de ces fichiers est directement proportionnelle au nombre total de nucléotides du motif décrit. Si nous lançons *MC-Search* avec le descripteur du motif tetraloop GNRA de la Table 2 et le fichier de structure PDB 1JJ2, nous obtenons après de nombreux calculs les fichiers de sortie suivants :

```
1JJ2_model_1-01.pdb
1JJ2_model_1-02.pdb
1JJ2_model_1-03.pdb
1JJ2_model_1-04.pdb
1JJ2_model_1-05.pdb
1JJ2_model_1-06.pdb
1JJ2_model_1-07.pdb
1JJ2_model_1-08.pdb
1JJ2_model_1-09.pdb
1JJ2_model_1-10.pdb
```

Table 3: Fichiers de sortie générés par *MC-Search* avec le descripteur du motif tetraloop GNRA énoncé à la Table 2 et le fichier de structure PDB 1JJ2

Ainsi, *MC-Search* a trouvé 10 instances du motif en question dans le fichier de structure PDB 1JJ2, chaque instance pouvant être visionné par un logiciel de visualisation 3D tel que Rasmol, comme illustré à la Figure 9.

Les limitations de *MC-Search*

Bien que l'application *MC-Search* soit efficace dans sa recherche de motifs et que plusieurs mises à jour ont été effectuées pour optimiser l'outil, il ne demeure pas moins que *MC-Search* comporte les deux limitations suivantes :

- *MC-Search* est une application de type stand-alone, ce qui 'effraie' et même décourage des utilisateurs potentiels qui auraient des connaissances limitées en informatique. Bien que son utilisation ne soit pas particulièrement difficile, il prend tout de même un certains temps pour un nouvel utilisateur de télécharger, installer et savoir exécuter avec les bons paramètres l'application. Si les outils 'stand-alone' sont de moins en moins populaire, les outils Web en revanche gagnent en popularité, en autres, pour leur simplicité d'utilisation.

- La recherche de motifs peut prendre beaucoup de temps, particulièrement lorsque la description du motif s'avère complexe, mais également parce qu'il existe des milliers de fichiers PDB contenant des structures d'ARN. Il devient alors répétitif de rechercher les mêmes motifs dans les mêmes structures, ce qui arrive fréquemment quand deux personnes ou plus étudient les mêmes motifs. *MC-Search* ne fait pas de suivi sur les résultats trouvés. L'existence d'une base de données pouvant contenir toutes les descriptions de motifs utilisés ainsi que les instances de motifs s'avère pertinente. Le prochain chapitre se consacre à un outil Web qui intègre des motifs dans une base de données.

Chapitre 3 : Intégration de motifs avec *MC-Map*

MC-Map est avant tout une application Web permettant l'utilisation plus conviviale de *MC-Search*. Ainsi, tout utilisateur souhaitant rechercher des motifs structuraux n'est plus obligé de suivre les étapes d'installation, de configuration et d'exécution d'une application stand-alone avec les paramètres appropriés. Évidemment, il sera impératif que l'utilisateur de *MC-Map* décrive ses motifs avec le vocabulaire *MC-Search*, tel que mentionné dans le chapitre précédent, et spécifie quels seront les fichiers structuraux PDB de recherche. Un exemple d'utilisation de *MC-Map* est donné au chapitre suivant.

Mais *MC-Map* ne se limite pas seulement à une version Web de *MC-Search*. Le projet est également un outil d'intégration de motifs structuraux. En effet, l'intégration de données est une préoccupation de plus en plus importante dans le domaine de la bioinformatique. En ce qui attrait des motifs structuraux d'ARN, il ne semble pas exister de site d'intégration à ce jour sinon le site de classification SCOR (13). *MC-Map* stocke en mémoire toutes les instances de motifs trouvés par *MC-Search* et fait un suivi de ce qui a été recherché jusqu'à ce jour. Par exemple, si nous avons un motif X avec une description donnée et qu'une recherche de ce motif dans *MC-Map* a été lancée sur les structures PDB 1JJ2 et 2AWB, *MC-Map* gardera en mémoire le nombre d'instances trouvées du motif X pour chacune des structures PDB, que le nombre d'instances soit nul ou non. Ainsi, les résultats sont gardés en mémoire pour référence ultérieure, ce qui évite de lancer la même recherche par après et peut économiser beaucoup de temps en bout de ligne. *MC-Map* stocke l'information des motifs structuraux grâce à l'utilisation d'une base de données de type MySQL (http://www.mysql.com/).

Le grand avantage de l'utilisation de bases de données est la possibilité d'appliquer des opérations sur l'information stockée via des requêtes. Ainsi, nous pouvons aisément lancer des recherches, trier, associer et même classifier les instances de motif si nous

utilisons les requêtes appropriées. Le chapitre 5 est consacré en entier à la base de données qu'utilise *MC-Map*.

En plus d'intégrer des instances de motifs structuraux, *MC-Map* permet de localiser les instances dans ce que nous appelons des cartes d'ARN. Les cartes d'ARN sont des schémas bidimensionnelles de chacune des chaînes d'ARN décrites dans les fichiers PDB. Lorsqu'une instance de motif se retrouve dans la chaîne d'ARN, une barre de couleur figure alors dans la carte pour indiquer la position de l'instance. Chacune de ces cartes contient également une légende pour indiquer la position des nucléotides. Un exemple est illustré à la Figure 12 :

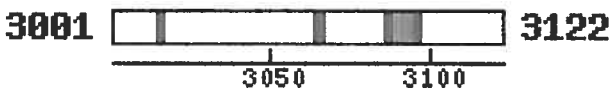


Figure 12: Carte d'ARN de la chaîne B du fichier PDB 1K8A, décrivant une structure co-cristallisée de Carbomycine A liée à la sous-unité ribosomale 50S de Haloarcula Marismortui. La chaîne B est ici représentée dans toute sa longueur dans un plan 2D, le premier nucléotide ayant le numéro 3001 et le dernier ayant le numéro 3122 dans l'ordre 5'-3'. Des instances de boucles internes (vert) ainsi que de tetraloop (rouge) sont ici retrouvées.

Les cartes d'ARN permettent de localiser chacune des instances de motifs étudiés et ainsi d'avoir une idée globale de la structure secondaire de chaînes d'ARN. Ces cartes peuvent avoir un rôle déterminant lorsque nous voulons faire de l'alignement multiple de plusieurs chaînes d'ARN du même type, comme par exemple les chaînes d'ARN de sous-unité ribosomales. Il est possible de cliquer sur chaque barre de couleur pour avoir davantage d'information sur l'instance de motif en question ou même de la visionner et de la télécharger. Cliquer sur une instance de motif affiche la petite fenêtre illustrée à la Figure 13 :

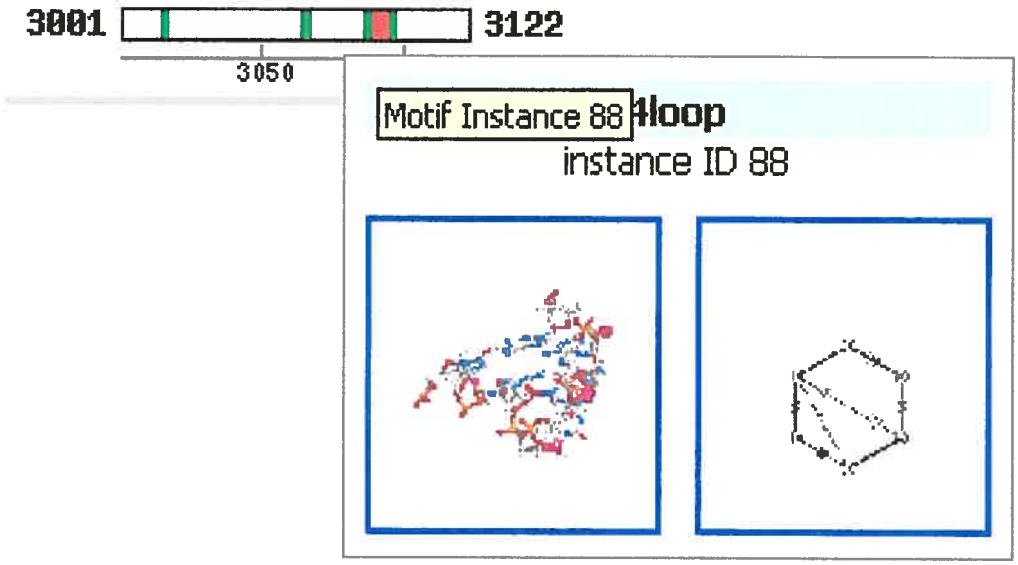


Figure 13: Fenêtre d'information apparaissant après avoir cliqué sur la barre rouge représentant une instance de motif tetraloop. La chaîne ici affichée correspond à la chaîne B de la structure PDB 1K8A, décrivant une structure co-cristallisée de Carbomycine A liée à la sous-unité ribosomale 50S de Haloarcula Marismortui.

La figure ci-dessus démontre un exemple de fenêtre d'information sur une instance de motif. Cette fenêtre apparaît suite au clic sur la barre rouge de la carte d'ARN représentant ici une instance de motif tetraloop. Deux images se retrouvent sur cette fenêtre, sous le texte affichant le type de motif ainsi que l'identificateur de l'instance de motif. L'image de gauche illustre un aperçu de l'instance tandis que l'image de droite illustre un graphe d'annotation de cette même instance.

À noter que pour l'instant, les cartes d'ARN n'offrent pas un affichage adéquat pour des instances de motifs multi brin ou se retrouvant sur plusieurs sections d'un même brin. En effet, une instance de motif peut s'étaler sur plus d'un brin d'ARN et/ou sections du même brin. Par exemple, à la Figure 12, les barres de couleur vertes (instance de motif de boucle interne) entourant la barre rouge (instance de motif tetraloop) font partie de la même

instance de motif. À ce jour, le seul moyen de vérifier si deux ou plusieurs barres de la même couleur représentent une même instance de motif est de comparer les identificateurs d'instance pour chacune de ces barres. Ceci étant donné que chaque instance de motif est assignée à un identificateur unique. À la Figure 13, nous pouvons observer que l'identificateur de l'instance de tetraloop est 88.

Finalement, la Figure 14 illustre l'architecture de *MC-Map* : nous pouvons constater que l'application Web se retrouve au sommet de l'architecture et traite directement avec la base de données MySQL. Cette dernière contient les informations sur les motifs définis, les structures PDB ainsi que les résultats de recherche de l'application *MC-Search*. Des exécutions de *MC-Search* sont lancées en arrière-plan, chacune d'elle recherchant un motif donné dans une structure PDB donnée. Les résultats obtenus viennent mettre à jour la base de données MySQL. À noter que plusieurs scripts entrent en jeux pour contrôler les exécutions de *MC-Search*, mais également pour insérer des données, tant alphanumériques que binaires, à la base de données. Ces scripts sont discutés au chapitre 6.

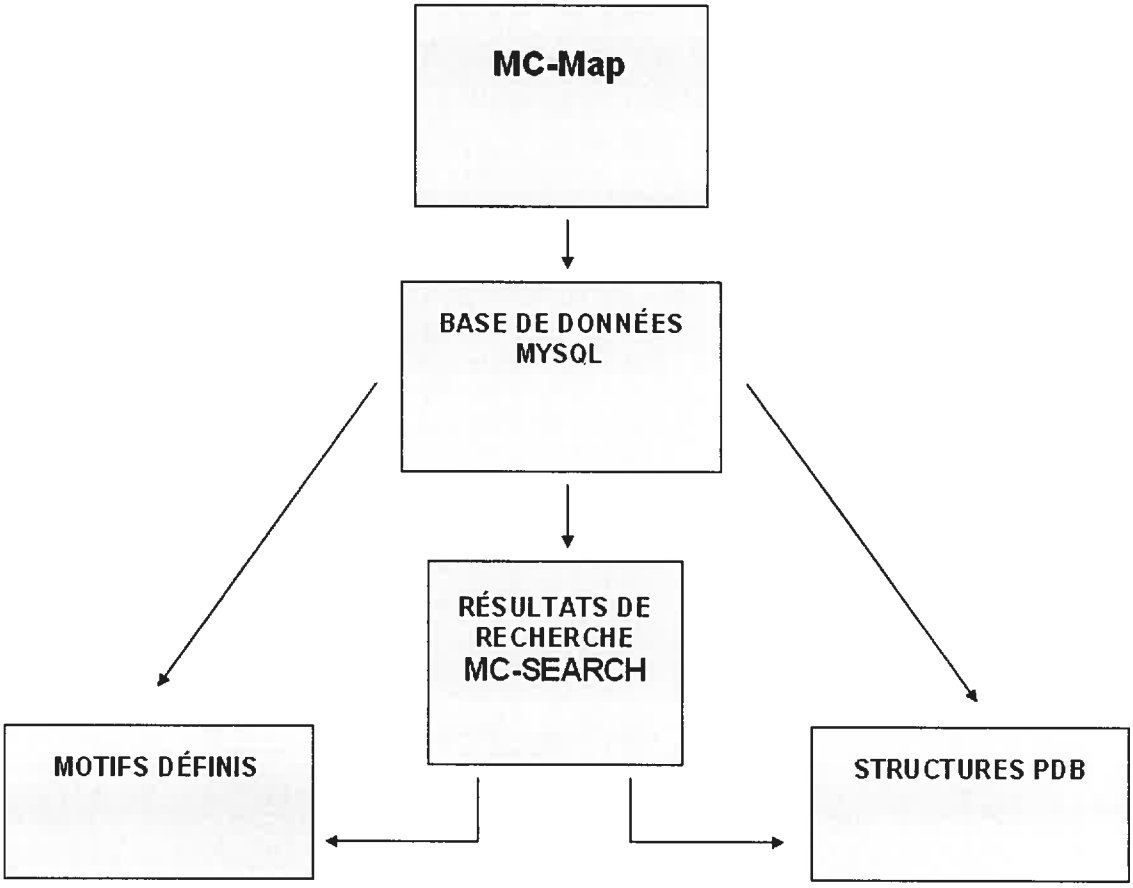


Figure 14: Architecture de *MC-Map*

Chapitre 4 : Article de publication

MC-Map: mapping RNA motifs to PDB structures

Nicolas St-Onge and François Major

Institute for Research in Immunology and Cancer
Department of Computer Science and Operations Research
Université de Montréal
PO Box 6128, Downtown station
Montréal QC H3C 3J7
Canada

Abstract

With the apparition of RNA macromolecules crystallographic models such as the ribosomal sub-units, RNA analysis and structural motif search has become an arduous task. Thus, some tools have been created for searching RNA structural motifs from these models. The application *MC-Search*, developed from our lab, enables the search of any user defined motif, given a description of this motif.

We believe that structural motif analysis requires an integration platform. Found motif instances need to be stored in such platform, so that they can be viewed and analysed later on by any user. The web application *MC-Map* has been implemented for mapping motif instances to PDB structures. Results are viewed in 2D graphical representation called RNA maps, revealing motif localization in RNA strands, as well as insights to the motif function. The web server can be accessed at http://www-lbit.iro.umontreal.ca/mcmap/.

Introduction

The 3D structure of an RNA strand greatly determines its function (1). Thus, efforts are being made to predict the 3D structure from the RNA sequence. One way to achieve it is by using a divide and conquer manner with the analysis of motifs. RNA motifs are recurrent RNA arrangements that have a similar shape within one or more molecules. The fact that these RNA segments are recurrent greatly suggests that they have a specific functionality within the molecule. But how can we efficiency search for RNA motifs within a molecule? One way is using a three-dimensional RNA pattern matching tool such as *MC-Search* (2).

Indeed, RNA structure can be described in terms of nucleotide interactions such as pairing, stacking and nucleotide adjacency. This list of nucleotide interactions can be assembled to form a complete RNA structural pattern, or in other words, an RNA motif description. *MC-Search* uses a nomenclature, based on Leontis & Westhof nomenclature (3) as well as Lemieux & Major nomenclature (4), to formally describe RNA structural patterns. The application maps the RNA structural pattern to its occurrences in predetermined 3-D structures (cf. by X-ray crystallography, NMR spectroscopy, etc.). The nomenclature is discussed in further details at the following URL:
http://www-lbit.iro.umontreal.ca/wiki/index.php/MC-Search_%28english_version%29

Although *MC-Search* can find three-dimensional RNA patterns with high accuracy, it is not very intuitive to visually map the location of each occurrence within the whole 3-D structure, especially if we deal with different RNA structural patterns to search. In addition, the fact that such pattern matching tool is used by different users and that resulting data is not collected into a single recipient such as a database, leads to possible loss of data.

MC-Map is a motif integration tool built on top of *MC-Search* and PDB (Protein Data Bank)(5) structures, mapping motif instances to PDB structures. Results are viewed in 2D graphical representation called RNA maps, revealing motif localization in RNA strands, as

well as insights to the motif function. The web server is found at http://www-lbit.iro.umontreal.ca/mcmap/.

Materials and Methods

MC-Search is a software application that has been created by the LBIT (http://www-lbit.iro.umontreal.ca) for searching in a PDB structure a given user defined RNA motif. *MC-Search* is currently only available on Linux operating system and can be built from *MC-Core* library (http://sourceforge.net/projects/mccore).

MC-Map is a web application created with PHP server-side programming language, client-side Javascript language and MySQL database. It runs *MC-Search* commands from the web site, without requiring the user to have prior how-to-use knowledge of *MC-Search*. Results are stored into the MySQL database and are afterward available for all users. The web interface has been designed for being the most user-friendly possible and many components have been added so that the user can easily and quickly make operations with motifs and PDB structures. Among these components are Softcomplex Tigra Javascript components (http://www.softcomplex.com). Map pictures are created dynamically using PHP GD graphical library.

In addition, some Bash Shell Unix scripts have been used for processing data files generated by *MC-Search* application. A modified version of *Rasmol* program (http://www.umass.edu/microbio/rasmol) is used for generating preview pictures for individual motifs while another external application is used to create secondary structure representations.

Results and Discussion

Results

We have created a database of all structural motifs we have been studying so far. *MC-Map* is a Web server to this database. The current PDB dataset is containing 507 structures that result mostly from X-Ray crystallography experiments (no NMR structure for now). Table 1 lists all these structures.

100D	1E7K	1H4S	1KNZ	1NTB	1RXA	1VQ9	1YJW	259D	2BX2	2G8H	405D
157D	1E7X	1H8J	1KQ2	1NUJ	1RXB	1VQK	1YRJ	280D	2C4Q	2G8I	409D
161D	1EC6	1HC8	1KUO	1NUV	1S03	1VQL	1YTU	283D	2C4Y	2G8K	413D
165D	1EFO	1HDW	1KUQ	1NYI	1S72	1VQM	1YTY	299D	2C4Z	2G8U	418D
168D	1EFW	1HE6	1KXK	1O0B	1S76	1VQN	1YVP	2A04	2C50	2G8V	419D
1A34	1EHZ	1HMH	1L2X	1O0C	1S77	1VQO	1YXP	2A0P	2C51	2G8W	420D
1A9N	1ET4	1HQ1	1L3D	1O3Z	1SA9	1VQP	1YYK	2A1R	2CKY	2G92	421D
1APG	1EUY	1HR2	1L3Z	1O9M	1SAQ	1VS5	1YYO	2A43	2CSX	2GCS	422D
1AQ3	1EVP	1HYS	1L8V	1OOA	1SDR	1VS6	1YYW	2A8V	2CT8	2GCV	429D
1AQ4	1EVV	1I2X	1L9A	1OSU	1SDS	1VS7	1YZ9	2AB4	2CV0	2GDI	430D
1ASY	1EXD	1I2Y	1LC4	1P79	1SER	1VS8	1Z43	2AGN	2CV1	2GIC	433D
1ASZ	1F1T	1I5L	1LNG	1PGL	1SI3	1VS9	1Z7F	2AKE	2CV2	2GIS	434D
1AV6	1F27	1I6U	1LNT	1PJG	1SJ3	1W55	1ZBH	2ANN	2D2K	2GJE	435D
1B23	1F7U	1I7J	1M5K	1PJO	1SJ4	1WMQ	1ZBI	2ANR	2D2L	2GO5	437D
1B2M	1F7V	1I9V	1M5O	1PVO	1SJF	1WNE	1ZBL	2AO5	2D6F	2GUN	438D
1B7F	1F7Y	1I9X	1M5P	1Q29	1T0D	1WPU	1ZCI	2ASB	2DB3	2GY9	439D
1BMV	1F8V	1ICG	1M5V	1Q2R	1T0E	1WRQ	1ZDH	2ATW	2DER	2GYA	462D
1BR3	1FEU	1ID9	1M8V	1Q81	1TFW	1WSU	1ZDI	2AWB	2DET	2GYB	464D
1C0A	1FFK	1IDW	1M8W	1Q82	1TN2	1WVD	1ZDJ	2AWE	2DEU	2GYC	466D
1C9S	1FFY	1IHA	1M8X	1Q86	1TRA	1X9C	1ZDK	2AZ0	2ESI	2HYI	468D
1CSL	1FG0	1IK5	1M8Y	1Q93	1TTT	1XJR	1ZE2	2AZ2	2ESJ	2IZN	469D
1CVJ	1FIX	1IL2	1M90	1Q96	1U0B	1XMQ	1ZEV	2AZX	2ET3	2J0Q	470D
1CX0	1FJG	1IVS	1MDG	1Q9A	1U8D	1XOK	1ZFR	2B2D	2ET4	2J0S	471D
1D4R	1FUF	1J1U	1MHK	1QA6	1U9S	1XP7	1ZFT	2B2E	2ET5	2TRA	472D
1D87	1FXL	1J6S	1MJI	1QBP	1UTD	1XPE	1ZFV	2B2G	2ET8	300D	479D

1D88	1G1X	1J7T	1MMS	1QC0	1UVI	1XPF	1ZFX	2B3J	2EZ6	301D	480D
1D96	1G2E	1J8G	1MSW	1QF6	1UVJ	1Y26	1ZH0	2B57	2F4S	310D	483D
1D9H	1G2J	1J9H	1MSY	1QLN	1UVK	1Y27	1ZH5	2B8R	2F4T	315D	485D
1DDL	1G4Q	1JB8	1MWL	1QRS	1UVL	1Y39	1ZHO	2B8S	2F4U	332D	4TNA
1DDY	1G59	1JBR	1MZP	1QRT	1UVM	1Y3O	1ZJW	2BBV	2F8K	333D	4TRA
1DFU	1GAX	1JBT	1N1H	1QRU	1UVN	1Y3S	1ZL3	2BCY	2FCX	353D	5MSF
1DI2	1GID	1JID	1N32	1QTQ	1VBX	1Y6S	1ZSE	2BCZ	2FCY	354D	6MSF
1DK1	1GKW	1JJ2	1N35	1QU2	1VBY	1Y6T	1ZX7	2BE0	2FCZ	359D	6TNA
1DNO	1GSG	1JZV	1N38	1QU3	1VBZ	1Y73	1ZZ5	2BEE	2FD0	361D	7MSF
1DNT	1GTF	1K8A	1N77	1QVG	1VC0	1Y95	1ZZZ	2BGG	2FGP	364D	
1DNX	1GTN	1K8W	1N78	1R3E	1VC6	1Y99	205D	2BH2	2FK6	377D	
1DQF	1GTR	1K9M	1N7A	1R3O	1VC7	1YFG	216D	2BJ6	2FMT	393D	
1DQH	1GTS	1KD1	1N7B	1RC7	1VFG	1YHQ	217D	2BNY	2FQN	394D	
1DRZ	1H2C	1KD3	1N8R	1RGA	1VQ4	1YI2	222D	2BQ5	2FZ2	397D	
1DUH	1H2D	1KD4	1NB7	1RLG	1VQ5	1YIJ	246D	2BS0	2G4B	398D	
1DUL	1H38	1KD5	1NJI	1RMV	1VQ6	1YIT	247D	2BS1	2G5K	3TRA	
1DUQ	1H3E	1KFO	1NLC	1RNA	1VQ7	1YJ9	248D	2BTE	2G5Q	402D	
1DZS	1H4Q	1KH6	1NTA	1RPU	1VQ8	1YJN	255D	2BU1	2G8F	404D	

Table 1: PDB dataset currently used in *MC-Map* database

Table 2 shows all motifs that are currently part of the database and that have been the object of study at the LBIT lab, as well as the total number of instances found for each motif in the current PDB dataset of *MC-Map*, listed in Table 1. This list is available at the URL http://www-lbit.iro.umontreal.ca/mcmap/motif.php?action=status&motifid=all. The table is spread on two columns for convenience reasons. On the website, icons are located next to each motif name, enabling any user to get related information about the given motif through a popup window. We believe that motifs with a total number of instances above 1000 are motifs with insufficient structural restrictions, while motifs with less than 10 instances are too restrictive.

MOTIF NAME	TOTAL INSTANCES
C-motif 3-1	59
C-motif 4-2	31
Heptaloop A (simple)	1377
Heptaloop B (W-C)	1009
Heptaloop C	163
Heptaloop D (3stack min)	297
Hexaloop A (simple)	1660
Hexaloop B	322
Hexaloop C (triloop + bulge)	48
Hexaloop D (4stack)	44
Iloop 1-0 (5nuc) A	736
Iloop 2-0 (6nuc) A	3792
Iloop 2-1 (7nuc) A	1523
Iloop 3-1 (8nuc) A *	313
Iloop 3-2 (9nuc) A *	2997
Kink turn (PPS)	336
Kink turn (PSS)	188

MOTIF NAME	TOTAL INSTANCES
Octaloop A (simple)	1535
Octaloop B	62
Pentaloop (Tetraloop Like)	187
Pentaloop A	642
Pentaloop B	396
Pentaloop C	221
Sarcin-ricin A *	78
Tetraloop A (simple)	1599
Tetraloop B (Watson-Crick)	985
Tetraloop GNRA	376
Tetraloop YNAG	52
Tetraloop YNCG	117
Triloop A (simple)	1515
Triloop B (canonical)	158

Table 2: Motifs that currently exists in *MC-Map* database with total number of instances found in the PDB dataset listed in Table 1. The * symbol denotes motifs that have not been searched yet in every PDB structure of the dataset, suggesting that the assigned total motif instances number might be larger.

In total, *MC-Map* has detected so far 22818 motif instances for 31 motifs in 507 PDB structures. These PDB structures are mostly resulting from X-Ray crystallography experiments with none resulting from NMR spectroscopy.

Discussion

The key aspects of *MC-Map* are the following:

Make *MC-Search* application more accessible for the biologist community

Downloading stand-alone applications, installing and learning how to use them are often time-consuming and sometimes painful for the common user who does not necessary have a broad knowledge of computer software. Potential users are thus hesitant to use stand-alone application. A web application is faster to preview and test, as it does not require any installation procedure. Although users may currently not launch *MC-Search* application online, no prior-knowledge of this application is required in order to view results.

MC-Search data storage

MC-Map stores *MC-Search* results into a shared database so that results can remain fully accessible afterward. This gives the advantage to avoid any user to search again a given motif if it has been searched previously. Indeed, it may take hours for *MC-Search* to find a particular motif in a large group of PDB structures.

Map motif results into a chain viewer

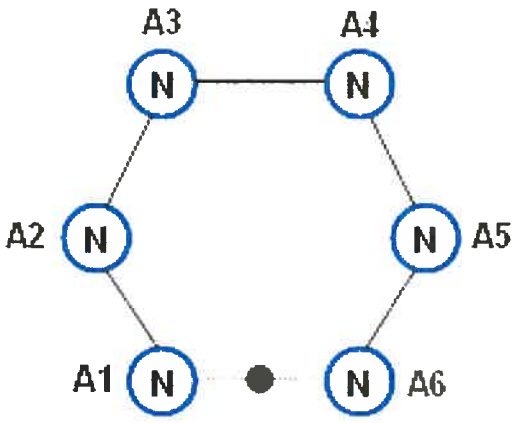
The chain viewer displays PDB chains containing motif of interests. It is particularly handy for localizing motif instances and thus determining distances with other motifs. This is especially interesting for data integration in that it allows users to have a global view on a PDB structure with all the motifs it contains and related information.

Easy data integration development

MC-Map has been built for easy data integration development, that is, it would be relatively easy to integrate new information or new tools to the web server. This could be achieved either by adding new database tables, adding web pages or linking the current data to other web server.

Program Example

Let's illustrate *MC-Map* functionality with an example. First, let us define a few motifs that will be mapped to the 16S and 23S ribosomal units. The description of a regular tetraloop could be as follows, using *MC-Search* motif description nomenclature:



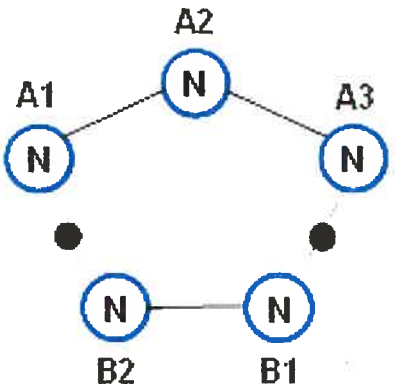

```
// Tetraloop with
// Watson-Crick
// flanking pair

sequence( RNA A1 NNNNNN )

relation(
A1 A6 { W/W }
)
```

Figure 1: On left is found a graph of the secondary structure of a tetraloop motif. Circles are showing nucleotides. The N within each circle denotes that the nucleotide can handle any value, that is, A,C,G or U. A1 to A6 are denoting nucleotide names that are used in the *MC-Search* descriptor file. Black lines are showing adjacency within nucleotides while the dashed line with a filled circle indicates a Watson-Crick base pair between the first and the last nucleotide of the 6-nucleotide strand. On left is found the descriptor script representing the same motif.

From the above description, one can figure out that the motif is single-stranded and six nucleotides long as shown on the sequence code line. In addition, the first nucleotide (referred as A1) and the last nucleotide (referred as A6)) are connected via a Watson-Crick pairing relation, denoted by the 'W/W' syntax within the curly braces. Relation properties are explained in further details on *MC-Search* Wiki page at the URL http://www-lbit.iro.umontreal.ca/wiki/index.php/MC-Search_%28english_version%29#relation. Let's add now an internal loop as follows:




```
// Internal loop 1-0 with
// Watson-Crick flanking
// pairs

sequence ( RNA A1 NNN )
sequence ( RNA B1 NN )

relation(
// iLoop flanking pairs
A1 B2 { W/W }
A3 B1 { W/W }
)
```

Figure 2: On left is found a graph of the secondary structure of an internal loop 1-0 motif. Circles are showing nucleotides. The N within each circle denotes that the nucleotide can handle any value, that is, A,C,G or U. A1 to A3 and B1-B2 are denoting nucleotide names that are used in the *MC-Search* descriptor file. Black lines are showing adjacency within nucleotides while the dashed lines with a filled circle indicate Watson-Crick base pair interactions at both ends of the 5-nucleotide motif. Note that this motif may contain up to two strands. On left is found the descriptor script representing the same motif

Internal loops motifs are structurally more complex to describe than hairpin loops (such as the tetraloop motif previously described), as one or two strands may be implied. This explains why two sequences are described at the top of the description. The above motif is called an internal loop 1-0, 1-0 standing for one nucleotide left unpaired on the first strand (A2) and none unpaired on the second strand. The internal loop motif is closed at both ends with Watson-Crick pairing interactions (nucleotide A1 pairing with nucleotide B2, nucleotide A3 pairing with nucleotide B1).

Although motif creation is currently not accessible to public, we assume that the given motifs are part of the *MC-Map* internal database. Any user can map a selected motif from the motif results page (see Table 2) to any PDB structure from the database (see Table 1). Results may take some time before appearing as the internal PDB structure list is quite exhaustive. In addition, it is not possible to view more than one motif at a time from the

motif results page. This is achieved however through a project in *MC-Map*. A project is essentially a selection of motifs and PDB structure that gives to one user a restricted view to *MC-Map* internal database. Figures 4 and 5 show views of such selections:

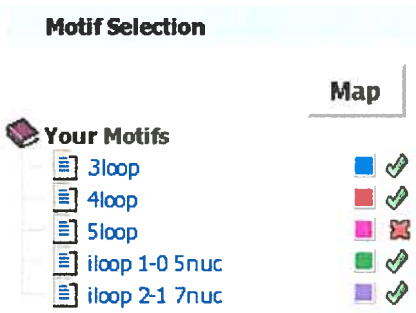


Figure 3: This motif selection contains five different motifs, each with an associated color and a status picture

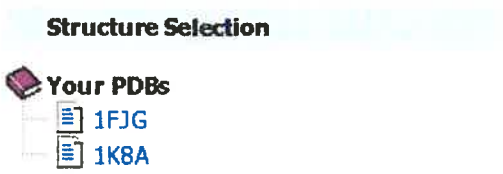


Figure 4: This structure selection contains two PDB structures, 1FJG and 1K8A. The first structure contains a 16S ribosomal RNA strand while the second one contains a 23S ribosomal RNA strand.

In the above project example, five motifs and two PDB structures have been selected. Our tetraloop and internal loop 1-0 motifs have been included in the motif selection. The triloop, pentaloop and internal loop 2-1 have been added. One can notice from Figure 3 that each motif selection is assigned a color box, as well as a status icon displaying either a passed or failed picture. The color box displays the motif assigned color, which by default is the dark blue color. Each motif is assigned a color in order to distinguish motifs on the RNA map images. Clicking on color boxes makes the palette color window popup for modifying the motif assigned color. The status icon next to the color box is representing the motif search status, either showing a passed or failed picture. A passed picture informs the user that the associated motif has been already searched in all PDB structure from the project structure selection, with or without results. Similarly, a failed picture denotes that the associated motif has not been searched yet in all selected structures, either that no search has been launched so far or that a current running searched is not yet completed. In

Figure 4, only the pentaloop motif has the incomplete search status. An incomplete status search affects the results in that resulting RNA map images may be missing information.

One can access the *MC-Search* menu by clicking on the status icons. Such menu displays the PDB structures for which a specific motif or a group of motifs have the incomplete search status. Although it is currently disabled to the public, one can find buttons for launching motif search on the menu as well. Links for viewing motif results are also present from the *MC-Search* menu. These motif results pages are similar to the one found on Figure 1, but are either restricted to one specific motif or the whole project motif selection.

Clicking on the Map button (see Figure 4) displays the RNA map images. These map images are 2-D representations of RNA strands found in PDB selected structures. Below are the RNA maps for the example motif and structure selection:

Figure 5: RNA map images for all selected motifs and PDB structures. Color bars are showing motif instances hits on each RNA strand. Colors are referring to motif associated colors shown on Figure 3 (e.g. a blue bar indicates the presence of a triloop motif instance, a pink bar indicates the presence of a pentaloop, etc.). The first RNA map shows a 16S rRNA strand found in 1FJG PDB structure while the two others show respectively the 23S and 5S rRNA strands from 1K8A PDB structure. The two first strands are typical from ribosomal subunit as they are more than 1500 nucleotides long. Numbers found on each strand size are respectively denoting the 5' and 3' residue numbers. Numbers below the map images are residue number delimiters.

Color bars from RNA map images denote motif instance hits. Colors refer to the motif associated colors. Clicking on any mapped motif instance displays the motif instance information window as shown on Figure 7. This window contains a preview picture of the motif instance as well as the annotation figure. Clicking on the first picture dynamically generates the PDB motif structure according to PDB format, so that we can either view the motif using a helper application such as Rasmol, or save it on the disk. The second picture reveals annotation information detected within the motif instance.

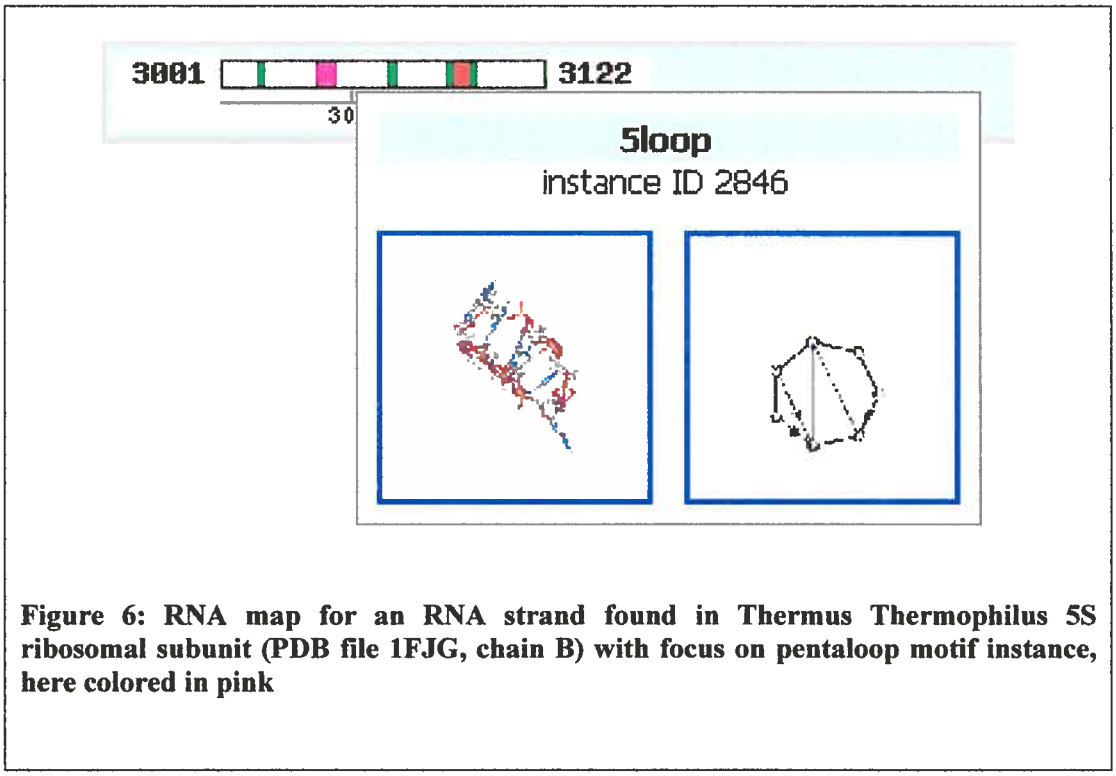


Figure 6: RNA map for an RNA strand found in Thermus Thermophilus 5S ribosomal subunit (PDB file 1FJG, chain B) with focus on pentaloop motif instance, here colored in pink

RNA map images allow a better comprehension of motifs within structures, especially regarding their function.

Supplementary Data

While it is possible to view all motif instances for a single motif over the whole PDB structure database without touching projects, *MC-Map* application is project oriented. That is, each user must create a new project or load an existing one in order to map several motifs in a restricted PDB structure selection. A project simply consists of a motif and a PDB structure selection. These settings help the user focusing on specific structural aspects. Once the project is set up, any user can access it and even modify it anytime later on. Note that motifs can be grouped in motif sets while PDB structures can be grouped in a dataset, which can ease data manipulation.

In order to get more familiar with *MC-Map* application, the following covers the different forms, menus and results pages of the application that one user may need throughout his project.

Creating a new project

MC-Map home page gives the user the option of either opening an already existing project or creating a new one. Creating a new project brings the user to the following page:

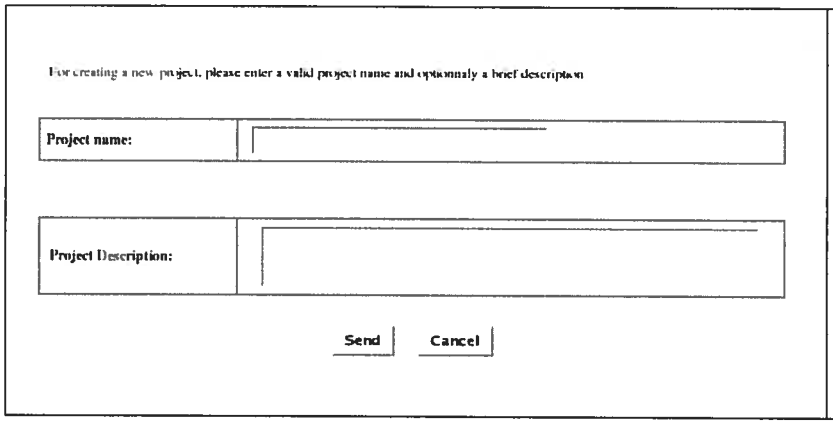


Figure 7: Creating a new project in *MC-Map*

On the 'Create new project' page, the user must fill up a form by providing a name and an optional short description to the project. The name must be unique to the project, otherwise

a message error shows up. Once the form has been successfully posted on the web, we are ready to set it up. Figure 9 shows a snapshot view of an empty project.

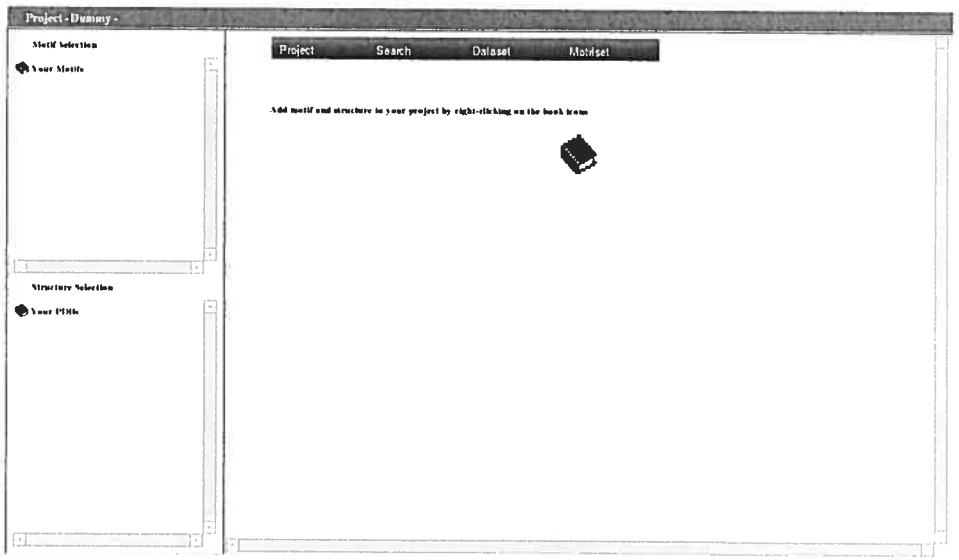


Figure 8: Setting up a new project in *MC-Map*

At the beginning of a new project, there are no motifs, nor any PDB structure selected. For adding motifs to the project, one user has either the options of adding a motif set, adding specific motifs or new motifs. Either case can be achieved with the motif contextual menu. This menu popup when the user right-clicks on the 'Your Motifs' text. Figure 10 shows a snapshot of the motif contextual menu.

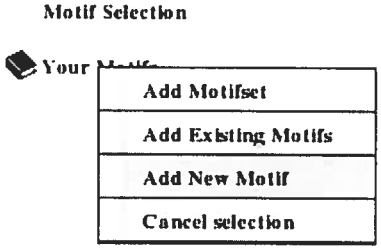


Figure 9: *MC-Map* motif contextual menu

Creating a new motif definition

New motifs are motifs that are not stored yet in the database. Adding new motifs is not an option currently accessible to the public as it requires eventual search calculation, which is for now too demanding for the web server CPUs. However, it is going to be discussed in the next lines in order to give an idea how motifs are created and inserted into the database. Selecting the 'Add New Motif' option from the motif contextual menu brings the 'New motif' form as shown in Figure 11, which is needed for creating a new motif.

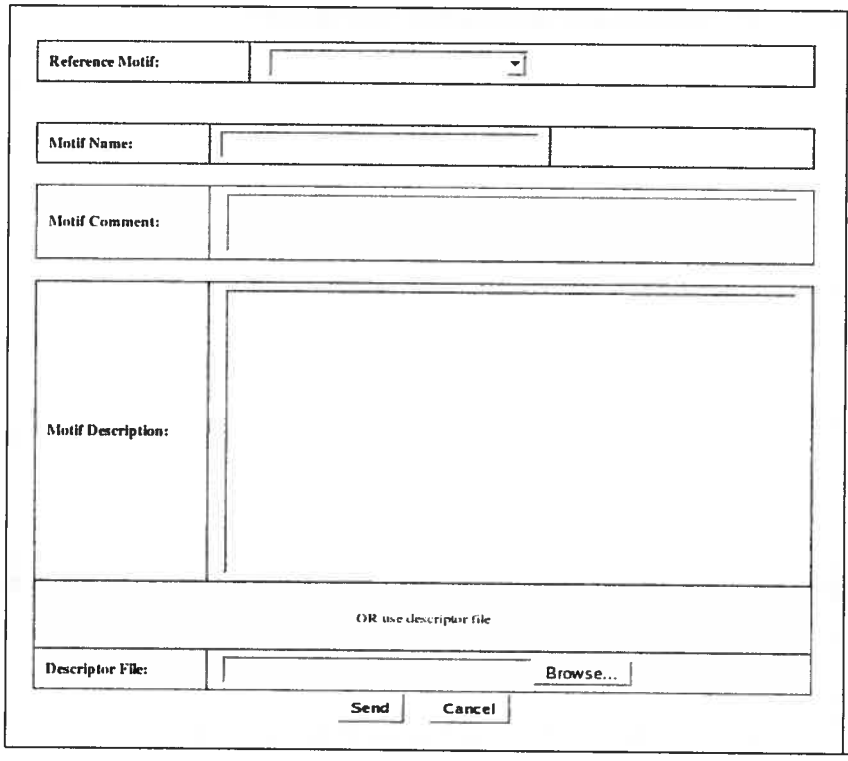


Figure 10: Creating a new motif in *MC-Map*

Each created motif needs a unique name, an optional comment as well as a motif description. This last parameter must follow the *MC-Search* description nomenclature as documented on *MC-Search* Wiki page. The motif description is either supplied by the user in the motif description text field or via text upload. If a description file is uploaded, then any text found in the motif description text field is ignored. Once the form is posted, motif

information is validated by *MC-Search* application. If the motif is successfully validated, then it is inserted into the database. Otherwise, motif corrections are required.

It is not possible to edit motif information once a motif is validated. However, one can define a new motif based on another one (a reference motif), as long as this one exists in the database. From the 'New motif' form, one selects the reference motif from the 'Reference motif' drop-down menu. Selecting any motif from this control fills up the entire form with the reference motif information. One can then edit any field and submit the new motif definition, as long as the motif name has been modified in order to avoid duplicated motif names.

Adding existing motifs and motif set

Adding existing motifs or a motif set is similar to adding a new motif in that one must select the appropriate option from the motif contextual menu. Either case brings the user to the appropriate form. Figure 12 shows a snapshot of the 'Add existing motif' form while figure 13 shows a snapshot of the 'Add motifset' form.

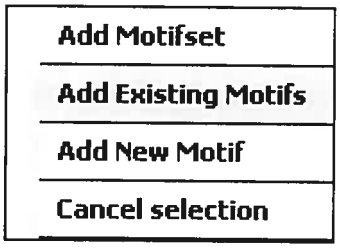


Figure 11: The motif contextual menu with the 'Add Existing Motifs' option selected

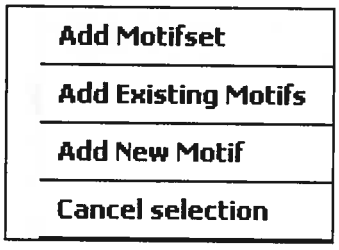


Figure 12: The motif contextual menu with the 'Add Motifset' option selected

The 'Add Existing Motifs' form displays a table listing all database motifs. Moving the cursor within the motif table highlights one motif at a time. Clicking on any motif adds the

selected motif to the project if not already part in the motif selection. Note that clicking on any motif icon does not add the associated motif to the project but rather popup a motif information window.

The 'Add Motifset' form displays a motifset drop-down menu that lists all motifset. Selecting any of the motifset displays a similar table to the one found on the 'Add Existing Motifs' form, containing all motifs part of the selected motifset. Clicking on the 'Add to Project' button adds the motifset to the project, if not already part in the motif selection. One can click on any of the listed motif of one motifset in order to individually add motifs.

Adding existing PDB structures and dataset

Adding existing PDB structures or dataset to the project is similar than adding motifs or motifset as seen previously. The only difference is that one needs the structure contextual menu in order to add structures, showing up when the user right-clicks on the 'Your Structures' text. The other steps are essentially the same.

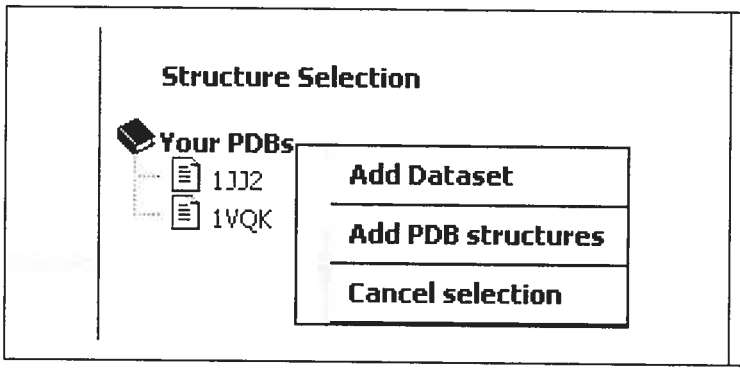


Figure 13: *MC-Map* structure contextual menu

Creating motif group and PDB structure group

One motif may be described several ways using *MC-Search* motif description nomenclature. Indeed, studying a particular motif using different motif description version

may proved to be pertinent. Thus, one user may wish to group all related motif description within the same group in order to view the different results. Similarly, one user may wish to map a specific motif or group of motifs to a group of PDB structures sharing similar properties, such as the source organism or organelle, the source experiment, the experiment resolution range, etc.

It is possible within *MC-Map* to create, modify and erase groups of motifs and groups of PDB structures. These options are currently not available to the public though. Note that groups of motifs are referred to motifset while groups of PDB structures are referred to dataset. Modifying dataset may be especially useful when new PDB structures are added to the structure database.

Exporting results

One can download all related motif instances from a project in either ZIP format or GZ format, in order to save results locally. The compress file contains all motif instances in PDB format. Each motif instance is found in a specific filename which prefix contains the original PDB filename, as well as an iteration number. PDB instances are grouped in directories representing each project motif. Exporting results is achieved using *MC-Map* "Project" drop-down menu and by selecting either one of the export options, as shown in Figure 14:

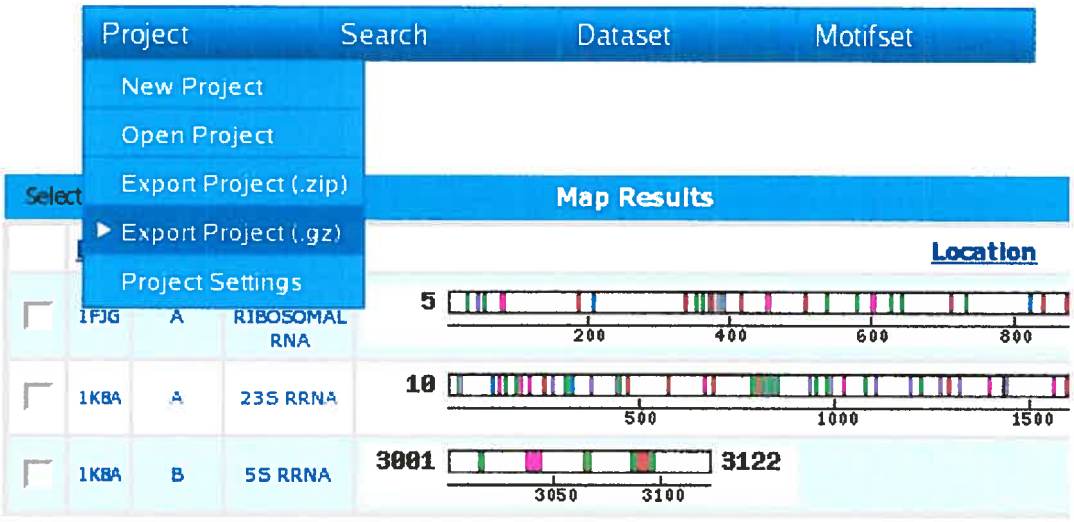


Figure 14: *MC-Map* "Project" drop-down menu has an option for exporting all motifs found in the project in either .gz format or .zip format

Conclusion

We have implemented a structural motif database listing the motifs we have been working on so far. This database shall grow in a short term perspective since we are continuously studying new RNA motifs and because the PDB database is growing as well. *MC-Map* web application allows motif data integration. PDB structure and motif instance information are stored online in a database. The main advantages of such approach are the possibilities of easily manipulating data through database request and easily integrating new data. So far, we can map motif instances to PDB structures using 2-D figures called RNA maps. In the future, we would like to extend the data integration either by storing new information (such as structure annotation data) in the database or linking *MC-Map* to other RNA web server applications. We would like as well to make *MC-Map* a downloadable application so that any user could analyze any defined motif locally.

Acknowledgments

The author would like to acknowledge his research director Francois Major for his help and assistance in the project. The author would also like to thank Martin Larose for his technical support, as well as Emmanuelle Permal for providing a PDB structure classification and Romain Rivière for providing its graphical tool for creating RNA graphs.

References

1. Saenger, W., *Principles of Nucleic Acid Structure*, Springer-Verlag, New York, USA, 1984

2. Larose M., Gendron P. et Major F., *MC-Search: a three dimensional RNA pattern matching tool*, RNA 2005 : Tenth annual meeting of the RNA SOCIETY, May 24-29, 2005

3. Leontis, N.B., Westhof, E., *Geometric nomenclature and classification of RNA base pairs*, RNA, 2001

4. Lemieux S. and Major F., RNA canonical and non-canonical base pairing types: a recognition method and complete repertoire, NAR, 30(19):4250-4263, (2002)

5. Berman, H.M., Westbrook, J., Feng, Z., Gilliland, G., Bhat, T.N., Weissig, H., Shindyalov, I.N. et Bourne, P.E., *The Protein Data Bank*, Nucleic Acids Res. 28. 235-242, 2000

Chapitre 5 : Base de données de MC-Map

La base de données utilisée par *MC-Map* est de type relationnel. Il a été choisi d'utiliser la base de données *MySQL* pour les raisons suivantes : sa gratuité (contrairement aux bases de données *Oracle* et *Microsoft Access*), sa simplicité d'utilisation et d'installation et également le fait qu'il s'interface facilement avec le langage de programmation *PHP*, qui a été conçu en ce sens. Ci-dessous se trouve l'architecture de la base de données :

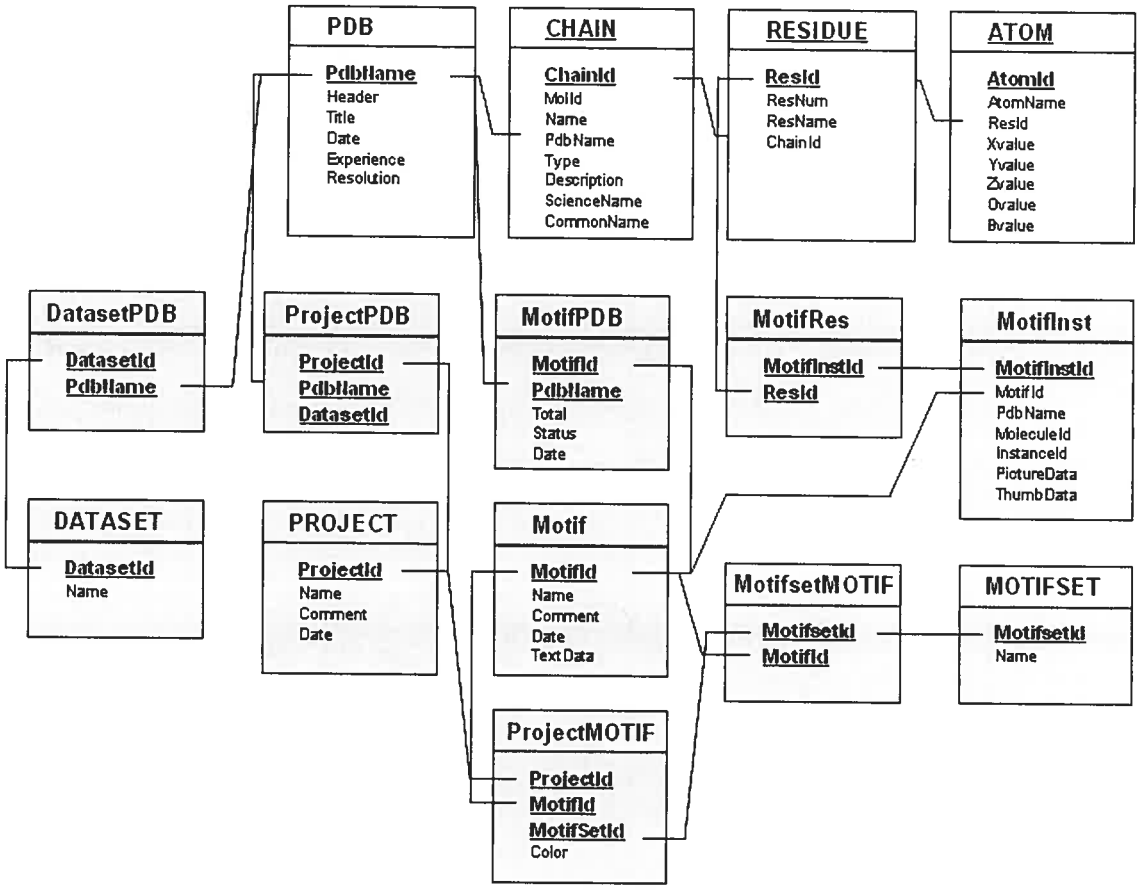


Figure 15: Architecture de la base de données de *MC-Map*

Chaque boîte illustrée représente une table de la base de données. Il y a en tout 15 tables dans la base de données de *MC-Map*. Le nom de la table est affiché en haut de la boîte, en gros caractères, tandis que les champs sont affichés en dessous, en plus petits caractères. La clé primaire de chaque table est représentée par l'ensemble des champs en caractère gras et soulignés. Enfin, les lignes qui relient les différentes tables viennent relier un champ identique appartenant à deux tables différentes, permettant ainsi de lancer des requêtes sur plusieurs tables à la fois. L'architecture globale tel qu'illustré à la Figure 15 pourrait être découpée en sections, nous permettant de mieux saisir le rôle de chaque table. Ces sections sont les suivantes : les tables de structure PDB, les tables de résultats de motifs, les tables de gestion de groupes de motif et structures PDB et finalement, les tables de gestion de projet.

Tables de structure PDB

Ci-dessous se trouvent les quatre tables impliquées dans l'information des structures PDB :

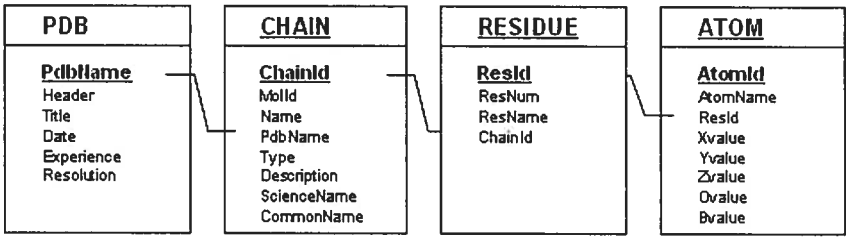


Figure 16: L'architecture des tables de structure PDB

Les tables 'Pdb', 'Chain', 'Residue' et 'Atom' dénotent l'architecture utilisée pour stocker l'information de structure PDB. Cette architecture vise à éviter au maximum la duplication d'information. Tel qu'illustré à la Figure 16, une structure PDB est implémentée comme un ensemble de chaînes peptidiques et/ou d'ARN, chacune de ces

chaînes étant une suite de résidus et finalement chacun de ces résidus étant un ensemble d'atomes. D'un point de vue base de données, nous pourrions décrire cet arrangement de tables comme suit : la table 'Pdb' contient la clé primaire 'PdbName' représentant le nom du fichier de la structure PDB, chaque nom de fichier étant unique à une structure PDB. Le champ 'PdbName' est également utilisé dans la table 'Chain', dont la clé primaire est le champ 'ChainId', permettant ainsi d'associer chaque chaîne à une structure PDB. Le champ 'ChainId' se retrouve à son tour dans la table 'Residue'. Ainsi, chaque résidu est associé à une chaîne qui elle-même est associé à une structure PDB. La table 'Residue' possède comme clé primaire le champ 'ResId', celui-ci se retrouvant dans une autre table, soit la table 'Atom'. Finalement, cette dernière table contient le champ 'AtomId' comme clé primaire, qui n'est toutefois pas utilisé dans toute autre table.

L'information stockée dans la table 'Pdb' se rapporte à la structure PDB : titre de la molécule, description, date de dépôt du fichier, la technique employée pour générer l'information de structure (le plus souvent cristallographie à rayons X) et finalement la résolution en Angstrom. Ces champs sont employés par l'interface *MC-Map* pour trier les structures PDB affichées en liste, tel qu'illustré à la Figure 17.

ALL PDB STRUCTURES				
PDB	HEADER	DATE	EXPERIMENT	RESOL.
100D	DNA/RNA CHIMERIC HYBRID DUPLEX	1994-12-05		1.90
157D	RIBONUCLEIC ACID	1994-02-01		1.80
161D	DNA/RNA	1994-02-10		1.90
165D	RNA/DNA	1994-03-21		1.55
168D	DNA/RNA	1994-04-08		2.00
1A34	COMPLEX (VIRUS/RNA)	1998-01-28	X-RAY DIFFRACTION	1.81
1A9N	COMPLEX (NUCLEAR PROTEIN/RNA)	1998-04-08	X-RAY DIFFRACTION	2.38
1APG	GLYCOSIDASE	1992-06-16		3.00
1AQ3	COMPLEX (COAT PROTEIN/RNA)	1997-08-06	X-RAY DIFFRACTION	2.80
1AQ4	COMPLEX (COAT PROTEIN/RNA)	1997-08-06	X-RAY DIFFRACTION	3.00
1ASY	COMPLEX (AMINOACYL-TRNA SYNTHASE/TRNA)	1995-01-19	SYNCHROTRON X-RAY DIFFRACTION 1A	0.00
1ASZ	COMPLEX (AMINOACYL-TRNA SYNTHASE/TRNA)	1995-01-19	SYNCHROTRON X-RAY DIFFRACTION 1A	0.00
1AV6	COMPLEX (TRANSFERASE/RNA)	1997-09-26	X-RAY DIFFRACTION	2.70
1B23	GENE REGULATION/RNA	1998-12-04	X-RAY DIFFRACTION	2.60
1B2M	HYDROLASE/RNA	1998-11-27	X-RAY DIFFRACTION	2.00
1B7F	RNA-BINDING PROTEIN/RNA	1999-01-23	X-RAY DIFFRACTION	2.60
1BMV	VIRUS	1989-10-09		3.00
1BR3	DNA/RNA	1998-08-13	X-RAY DIFFRACTION	3.00
1C0A	LIGASE/RNA	1999-07-15	X-RAY DIFFRACTION	2.40
1C9S	RNA BINDING PROTEIN/RNA	1999-08-03	X-RAY DIFFRACTION	1.90

Figure 17: Liste partiel des structures PDB de la base de données. Les champs au sommet de la liste peuvent être cliqués pour afficher la liste triée selon la sélection.

La table 'Chain' comporte l'information de chaque chaîne des structures PDB stockées dans la base de données. Le champ 'Type' indique s'il s'agit d'une chaîne d'ARN ou peptidique. Les champs 'Description', 'ScienceName' et 'CommonName' viennent renseigner sur la nature de la chaîne et de quel organisme la molécule appartient. À noter qu'une structure PDB peut contenir des molécules d'ARN ou de protéines provenant de plusieurs organismes (e.g. virus et cellule hôte). Présentement, bien que toute l'information sur les chaîne peptidiques, résidus et atomes se retrouvent dans la base de données, uniquement l'information sur l'ARN est utilisée et affichée.

La table 'Residue' contient l'information du nom et numéro de chaque résidu tel que figuré dans le fichier PDB. Le champ de la clé primaire, soit le champ 'ResId', est référé dans la table dans la table 'MotifRes' pour stocker l'information des résidus de chaque instance de motif. Nous verrons cette dernière table un peu plus loin dans ce chapitre.

La table 'Atom' contient toute l'information sur chacun des atomes, entre autres la nature de l'atome (O, C, N, etc.) mais également les coordonnées X,Y et Z. Le contenu des tables 'Residue' et 'Atom' est grandement utilisé pour la génération dynamique de fichiers PDB tel que discuté dans le chapitre précédent.

Finalement, les valeurs numériques des champs 'ChainId', 'ResId' et 'AtomId' doivent être non nulles.

Tables de résultats de motifs

Au centre de la Figure 11 se retrouvent les tables associées aux résultats de recherche de motif, soit les tables 'Motif', 'MotifInst', 'MotifRes' et 'MotifPdb'.

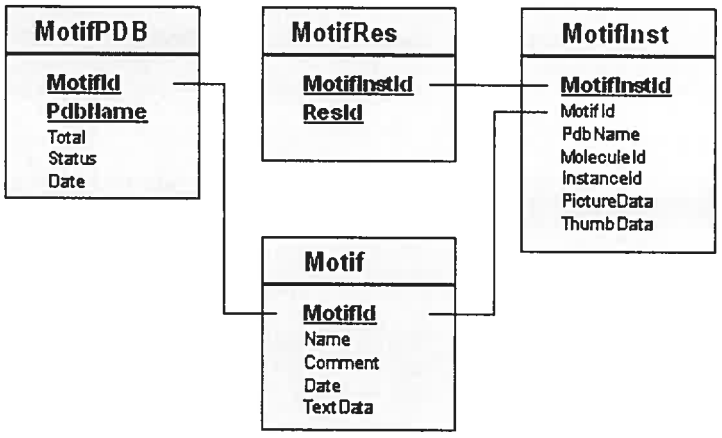


Figure 18: L'architecture des tables de résultats de motif

La table 'Motif' se remplie suite à l'insertion d'un nouveau motif (voir chapitre précédent, section 'Création de nouveaux motifs'). La clé primaire est le champ 'MotifId' dont la valeur est toujours non nulle. Cette table contient le champ 'Name' dont la valeur doit être unique, ainsi évitant la confusion entre deux motifs qui pourraient avoir un descripteur très semblable. Le champ 'Comment' contient le commentaire optionnel de l'auteur. Le champ de date réfère à la date ou le motif a été intégré à la base de données. Le champ 'Path' réfère au répertoire local où le motif a été sauvegardé avant d'être intégré à la base de données. Finalement, le champ 'TextData' contient le contenu du descripteur qui se fait valider par l'application *MC-Search* avant que le motif ne soit intégré. Cette description de motif sera plus tard utilisée pour lancer des recherches *MC-Search*.

La table 'MotifPdb' est une table de statistiques. La clé primaire de cette table est la combinaison des champs 'MotifId' et 'PdbName', le premier champ provenant de la table 'Motif' et le second, de la table 'Pdb'. Nous souhaitons ici faire le suivi de recherche pour chaque motif dans chaque structure PDB. Ainsi, retrouvons-nous les champs 'Total', 'Status' et 'Date'. Le champ 'Total' reflète le nombre d'instances de motif trouvées. Le champ 'Status' informe sur l'état de recherche *MC-Search*. Trois valeurs sont possibles pour ce dernier champ : 0 pour indiquer qu'aucune recherche n'a été lancé jusqu'à ce jour, 1 pour indiquer qu'une recherche est présentement en cours et 2, pour indiquer qu'une recherche a déjà été effectuée. Une fois que le statut est assigné à cette dernière valeur pour un motif donné et une structure PDB spécifique, il n'est plus possible de lancer la même recherche de nouveau pour ce même motif et cette même structure PDB. Cela évite de rechercher plus d'une fois la même chose. Par défaut, lorsqu'un nouveau motif est inséré dans la base de données, une ligne de données vient s'ajouter dans la table 'MotifPdb' pour chacune des structures PDB existantes, avec la valeur 0 pour le champ 'Status'. Ainsi, si le nouveau motif M vient s'ajouter dans la table 'Motif' et qu'il y a 100 structures PDB, 100 nouvelles lignes seront insérées dans la table 'MotifPdb' avec la valeur 0 pour le champ 'Status'.

La table 'MotifInst' se remplie suite à l'intégration de données par l'outil *pdb2db* (voir chapitre 6), après qu'une recherche *MC-Search* ait été lancée. La table 'MotifInst' a pour clé primaire le champ 'MotifInstId'. Cette table contient toute l'information sur une instance de motif précise : nature du motif, structure PDB d'origine, images d'aperçu, etc. Les champs 'PdbName' et 'MotifId' indiquent dans quelle structure le motif en question a été trouvé. Les champs 'MoleculeId' et 'InstanceId' indiquent l'ordre d'insertion des instances de motif pour une structure PDB et un motif donné. Ces champs réfèrent à la nomenclature *MC-Search* de génération de fichiers de sortie. Enfin, les champs 'PictureData' et 'ThumbData' réfèrent aux images d'aperçu générées par l'application *Rasmol* pour chacune des instances de motif. Tandis que le premier champ stocke le contenu binaire de l'image d'origine, le deuxième champ stocke le contenu d'une plus petite image, de la taille d'une grosse icône. Ceci a été pensé pour que l'affichage Web d'une liste d'instances de motif avec des images d'aperçu soit nettement plus rapide. Si l'usager désire observer l'instance de motif en plus gros format, il n'a qu'à cliquer l'icône en question, tel qu'illustré à la Figure 19.

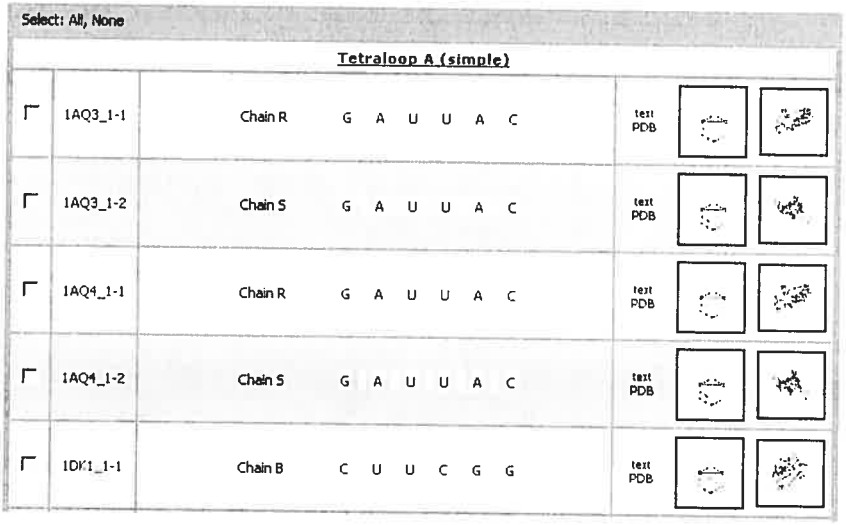


Figure 19: Liste partiel des instances du motif 'Tetraloop A'. Le nom de la structure PDB d'origine, de la chaîne d'origine et de la séquence complète de l'instance sont affichées. À droite se trouvent les images de graphe d'annotation et d'aperçu pour chaque instance.

La table 'MotifRes' associe à une instance de motif précis tous les résidus qui la constituent. Comme chacun des résidus est défini dans la table 'Residue', il suffit de référer au champ de clé primaire de cette dernière table pour en extraire toute l'information. L'élégance de cet approche est que l'information des résidus n'est pas dupliqué dans la table 'Residue' et la table 'MotifRes'. Ainsi, une fois que les structures PDB d'intérêt sont insérées dans la base de données, nous pouvons penser que la taille de la base de données, en terme d'espace de disque dur, n'évoluera guère par après, même si nous souhaitons insérer un très grand nombre de motifs et qu'il y a une possibilité que *MC-Search* trouve une multitude d'instances de motif. Ceci s'explique par le fait que la quantité de données de la base de données reliées aux motifs est négligeable par rapport à celle reliée aux structures PDB.

Tables de gestion de groupes de motif et structures PDB

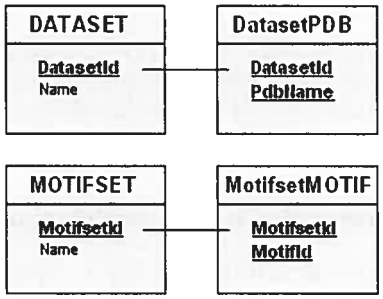


Figure 20: L'architecture des tables de gestion de groupes de motif et structures PDB

Les tables 'Dataset' et 'DatasetPdb' gèrent les groupes de structures PDB. Ces groupes sont crées par l'usager Web en cochant les structures PDB désirées à partir de la liste complète de structures PDB (voir chapitre précédent, section 'Création de groupes de motifs et de groupes de structures PDB'). La table 'Dataset' contient les champs 'Name' et 'DatasetId'. Malgré que ce dernier champ soit la clé primaire de la table, la valeur du

champ 'Name' se doit d'être unique pour éviter toute confusion entre les groupes de structures. La table 'DatasetPdb' vient associer un groupe de structures à plusieurs structures PDB. Les champs utilisés sont 'DatasetId' ainsi que 'PdbName', le premier provenant de la table 'Dataset' et le second, de la table 'Pdb'.

Les table 'Motifset' et 'MotifsetMotif' gèrent les groupes de motifs. L'architecture est similaire aux deux tables précédentes, mis à part le fait que l'on gère ici des motifs à la place de structures PDB. Notons que le champ 'Name' de la table 'Motifset' doit comporter également des valeurs uniques pour éviter toute confusions entre groupes de motifs. La table 'MotifsetMotif' utilise les champs 'MotifsetId' et 'MotifId', le premier champ provenant de la table 'Motifset' et le second, de la table 'Motif'.

Finalement, les valeurs numériques pour les champs 'DatasetId' et 'MotifsetId' doivent être non nulles.

Tables de gestion de projet

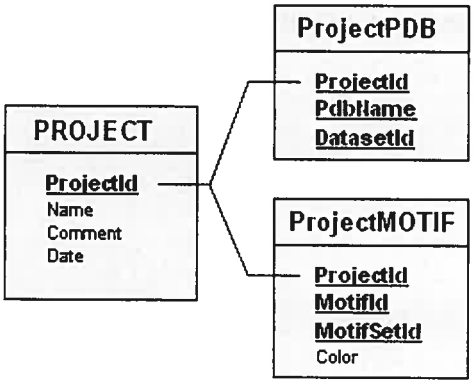


Figure 21: L'architecture des tables de gestion de projets

Trois tables gèrent les projets dans *MC-Map,* soit les tables 'Project', 'ProjectPdb' et 'ProjectMotif'. La table 'Project' se remplie suite à la création d'un nouveau projet par

un utilisateur Web (voir chapitre précédent, section 'Démarrage d'un projet'). La table 'Project' contient comme clé primaire le champ 'ProjectId'. Le champ 'Name' contient le nom du projet et sa valeur doit être unique pour éviter toute confusion entre les différents projets. Le champ 'Comment' contient le commentaire optionnel de l'auteur tandis que le champ 'Date' est la date à lequel le projet a été inséré dans la base de données.

On retrouve dans la table 'ProjectPdb' les champs 'ProjectId', 'PdbName' et 'DatasetId', tous les trois formant la clé primaire. En effet, chaque projet peut contenir des structures PDB non associées à des groupes (dites indépendantes), ainsi que des groupes de structures PDB. Lorsque nous insérons une structure PDB indépendante, une nouvelle ligne de données vient s'ajouter à la table 'ProjectPdb', le champ 'ProjectId' contenant l'identificateur du projet en cours, le champ 'PdbName' contenant le nom de la structure Pdb et le champ 'DatasetId' contenant la valeur 0. Cette dernière valeur indique que nous référons ici à une structure PDB et non à un groupe de structures, l'identificateur 'DatasetId' étant toujours non nul pour référer à un ensemble de structures PDB. Lorsque nous insérons un groupe de structures au projet, une nouvelle ligne de données s'ajoute à la table 'ProjectPdb' avec la même valeur pour le champ 'ProjectId', une valeur vide pour le champ 'PdbName' et la valeur numérique associée au groupe de structures à insérer pour le champ 'DatasetId'.

L'architecture de la table 'ProjectMotif' suit la même idée que celle de la table 'ProjectPdb'. L'idée de la table 'ProjectMotif' est d'associer au projet en cours des motifs indépendants et/ou des groupes de motifs. Les champs 'ProjectId', 'MotifId' et 'MotifsetId' forment la clé primaire. De nouveau, lorsque nous insérons un motif indépendant au projet en cours, une nouvelle ligne de données est insérée à la table, cette ligne contenant la valeur numérique du motif inséré ainsi que la valeur 0 pour le champ 'MotifsetId' et vice-versa lorsque nous insérons un groupe de motifs. La table contient en plus le champ 'Color' qui associe à un motif ou groupe de motifs une couleur afin de le distinguer des autres motifs ou groupe de motifs sélectionnés.

Architecture MySQL de la base de données

Pour plus d'informations sur l'architecture de la base de données utilisée par *MC-Map*, le fichier de source *MySQL* se trouve à l'Annexe A.

Chapitre 6 : Les modules et les scripts de *MC-Map*

Le chapitre qui suit est entièrement consacré aux modules et aux scripts du serveur *MC-Map*. Les modules, pour la plupart codés en langage PHP ou Javascript, sont les fichiers nécessaires à l'affichage de l'interface Web. Les scripts, codés en PHP ou langage *bash*, gèrent en arrière-plan les exécutions de *MC-Search* ainsi que l'intégration des données que ce dernier génère, en plus de l'intégration des fichiers de structure et des motifs définis.

Les modules de *MC-Map*

Les modules de *MC-Map* ont été classés dans la table suivante par catégorie. L'ensemble des modules totalise plus de 30 fichiers, dépasse les 20000 lignes de code ou 500K d'espace. Notons que les fichiers d'extension '.php' sont des modules PHP, les fichiers d'extension '.inc.php' sont des librairies de fonctions PHP et les fichiers d'extension '.js' des modules Javascript. Les modules de *MC-Map* se retrouvent pour la plupart dans le répertoire racine de l'application, mais également dans le sous-répertoire 'tigra', qui contient toutes les librairies Javascript permettant à l'usager de mieux contrôler l'environnement de *MC-Map*.

NOMS DE FICHIER	DESCRIPTION
dataset.php dataset.inc.php	Module pour afficher et gérer les groupes de structure PDB
dbase.inc.php	Librairie de fonctions de requêtes utiles avec la base de données MySQL de *MC-Map*

NOMS DE FICHIER	DESCRIPTION
download.php download.inc.php	Modules pour exporter les instances de motif du projet en cours
functions.js	Librairie de fonctions client Javascript. Ce module s'occupe entre autres de la mise à jour de l'information sur les différents cadres sur la page Web
index.php	Page de départ affichant les différents cadres du site Web (cadre de sélection de motifs, cadre de sélection de structures et cadre principal)
mapImage.php	Module pour créer dynamiquement l'image d'une carte d'ARN.
map.php map.inc.php	Modules permettant d'afficher une ou plusieurs cartes d'ARN selon la sélection de l'utilisateur
motif.php motifObj.php motif.inc.php motifset.php	Modules pour afficher et gérer l'information des groupes de motif. Le fichier 'motifObj.php' représente la classe Motif, utilisée lorsque l'information d'un motif donné est demandé.
mysql.inc.php	Fichier de configuration de *MC-Map*, principalement à l'accès à la base de données MySQL
pdbgen.php	Module de création dynamique de fichier PDB, utilisé pour visionner une instance de motif sélectionnée

NOMS DE FICHIER	DESCRIPTION
pdbparse.inc.php	Module nécessaire à l'utilitaire *pdb2db* afin d'extraire l'information des fichiers PDB et de lancer des requêtes *MC-Search* en chaîne
project.php project.inc.php	Module de gestion de projet, principalement utilisé lors de la création ou l'ouverture d'un projet
residueObj.php sequence.php sequenceObj.php sequence.inc.php	Module pour afficher en liste les instances d'un motif donné : séquences, figures d'annotation, figures d'aperçu, etc. Le fichier 'sequenceObj.php' représente la classe Séquence, qui elle-même nécessite la classe Residue, retrouvée dans le fichier 'residueObj.php'
session.php session.inc.php	Modules de gestion de l'information de session
barMenu.inc.php form.inc.php listMenu.inc.php table.inc.php	Modules utilisés pour l'esthétisme et les contrôles Web de *MC-Map*. Le fichier 'form.inc.php' est une libraire de fonctions couramment utilisées sur la plupart des pages Web de *MC-Map* : fonctions pour l'affichage des en-têtes et pied de tête, fonctions pour l'extraction des entrées de formulaire, etc. Le fichier 'table.inc.php' est une librairie de fonctions pour afficher divers tableaux : tableau d'affichage de motifs, tableau d'affichage de structure, tableau pour afficher un message, tableau de sélection de couleur de motif, etc. Les fichiers 'barMenu.inc.php' et 'listMenu.inc.php' affichent entre autres, les menus déroulants.

NOMS DE FICHIER	DESCRIPTION
TIGRA	Les librairies TIGRA sont principalement du code Javascript permettant à l'usager de mieux contrôler l'environnement de *MC-Map* : palette de couleur, tableaux de sélection, menu en arbres, menu déroulant principal, etc.
tree.php	Module pour afficher les arbres de sélection de motifs et de structures
view.php	Module pour visionner l'information d'un motif particulier : description du motif, affichage de l'image d'aperçu, affichage de la figure d'annotation, etc.

Table 4: Les modules de *MC-Map*, classés par catégorie. Sur la colonne de droite se retrouvent les noms de fichier regroupés par module, tandis que sur la colonne de droite se retrouve la description de chacun de ces modules.

Les scripts de *MC-Map*

MC-Map gère le stockage des données de structure dans les tables de données tel que décrites dans le chapitre précédent. Cette information contient entres autres l'information des fichiers de structure PDB, les fichiers descripteur de motifs, les instances de motifs trouvés, etc. Pour intégrer ces données à la base de données, *MC-Map* recourt à un script dit d'intégration. Le script d'intégration est codé en langage PHP et est nommé *pdb2db.php* ou *pdb2db*. Ce script recourt à d'autres scripts pour compléter sa tâche.

Le script *pdb2db* intègre les fichiers de structure PDB, les instances de motif générées par *MC-Search* ainsi que les icônes de ces instances. Pour intégrer les fichiers de structure PDB, *pdb2db* scanne ligne par ligne le contenu du fichier afin d'extraire les

données pertinentes sur la molécule décrite en général, sur chacune de ses chaînes d'ARN et de protéines, sur chacun des nucléotides et finalement sur chacun des atomes. Une fois ces données intégrées, *MC-Map* n'a plus besoin du fichier de structure PDB pour afficher les cartes d'ARN. Les instances de motif sont intégrées d'une manière différente. Rappelons que ces instances, trouvées par l'application *MC-Search*, sont générées en format PDB comme le fichier source. Le script p*db2db* scanne chaque ligne de ces fichiers d'instance, mais ne retire pas l'information sur chacun des nucléotides et atomes comme lors de l'intégration d'un fichier PDB en entier. Il vérifie plutôt si les nucléotides de l'instance de motif existent dans la base de données et, dans l'affirmatif, insère des pointeurs de ces nucléotides dans la table d'instance de motifs. Ceci a pour effet de minimiser l'espace d'entreposage requis pour les instances de motif. À noter qu'une fois que les instances d'un motif particulier ont été intégrées dans la base de données pour un fichier de structure PDB donné, il n'est plus possible d'intégrer d'autres instances pour ce même motif et fichier de structure par après. Ceci pour éviter une duplication des données, mais également pour avertir un utilisateur qu'il n'est plus nécessaire de lancer cette recherche à nouveau. Finalement, les icônes de motif sont intégrés dans la base de données comme de pures données binaires. Pour plus d'information sur l'utilisation du script *pdb2db*, des exemples sont décrits à l'annexe B.

Le script *pdb2db* utilise un autre script nommé C*lustermanager* qui gère des commandes de type 'cluster'. Ces commandes, disponibles pour l'instant que sur le réseau du DIRO, servent à lancer des applications gourmandes en ressources de processeur et mémoire, tel que *MC-Search*, sur des machines dédiées à cet effet. Ainsi, lorsque nous désirons rechercher un ou plusieurs motifs d'intérêt sur plus d'une structure PDB, *Clustermanager* partage les exécutions de *MC-Search* sur les machines 'cluster' disponibles. L'application *Clustermanager* s'avère très utile pour diminuer le temps d'exécution de plusieurs commandes *MC-Search* en même temps, mais ne peut certes pas diminuer le temps d'exécution d'une seule commande *MC-Search*, qui n'est pas une application exploitant plusieurs processeurs.

Le script *pdb2db* utilise également, lors de recherche d'instances de motifs, un script pour créer des images d'aperçu de chacune des instances trouvées ainsi qu'un script pour générer des illustrations d'annotation. Le script *pdbimage_all* vérifie l'existence de fichiers PDB dans un répertoire spécifié et, dans l'affirmative, crée des images d'aperçus pour chaque fichier PDB dans un répertoire de sortie. De même, le script p*db2jpg_all* vérifie l'existence de fichiers PDB dans un répertoire spécifié et, dans l'affirmative, crée les illustrations d'annotation.

Chapitre 7 : Perspectives futures

Le projet *MC-Map* est un projet Web d'intégration de données. Il a été conçu pour permettre aisément l'ajout de modules de code. Nous pourrions y apporter quelques options supplémentaires. Ces ajouts plausibles, énumérés ci-dessous, se retrouvent en deux catégories, soit dans un volet de perspectives à court terme ou dans un volet de perspectives à long terme, dépendamment du temps que ces ajouts nécessitent et de leur priorité :

Perspectives à court terme

Premièrement, comme mentionné au chapitre 3, *MC-Map* n'offre pas un affichage adéquat pour les instances de motifs multi-brin ou se retrouvant sur plusieurs sections d'un même brin. Un affichage devrait être envisagé pour renseigner l'usager sur la localisation des différentes sections d'une même instance.

Également, il y aurait une grande utilité à ce que l'application puisse être déployé localement sur les machines au lieu de résider uniquement sur le Web. En effet, si plusieurs recherches de motifs devaient s'effectuer simultanément, les ressources du serveur Web s'avèreraient insuffisantes, d'où le grand avantage à ce que l'application soit exécutée localement sur les machines des utilisateurs. Ainsi, chaque utilisateur pourrait lancer des recherches de motifs en fonction des limites de leur machine. Un autre avantage à cette option serait la confidentialité assurée des résultats : *MC-Map* est présentement configurée à ce que tous les résultats soient accessibles au grand public, les structures PDB comme les motifs. Il serait toutefois souhaitable dans une telle éventualité que les différents utilisateurs 'locaux' de *MC-Map* puissent avoir l'option d'échanger leurs résultats. S'il est présentement possible d'exporter des résultats, ainsi faudrait-il pouvoir importer des résultats.

Le fait que les données de *MC-Map* soient stockées dans une base de données donne beaucoup de possibilités à un usager de lancer des requêtes SQL. Une option intéressante serait un système permettant de classifier des résultats selon une métrique définie par l'utilisateur : séquence précise, nombre de chaînes contenues dans le motif, éléments de structure secondaire si nous avons accès à de l'information d'annotation, etc. Une option d'impression ou de sauvegarde dans un fichier serait alors disponible, permettant à l'utilisateur de revenir plus tard sur ses résultats de classification.

Perspectives à long terme

Il aurait été pertinent d'intégrer à *MC-Map* l'information d'annotation des structures PDB que l'application *MC-Search* utilise afin de trouver un motif d'intérêt. Ainsi, cela permettrait d'étudier plus en détails les résultats de recherche de motifs et même de classifier ces résultats en fonction de critère de structure secondaire (empilements, appariements, etc.). Cet ajout nécessiterait toutefois un certains temps à implémenter en raison de la complexité à intégrer l'information d'annotation dans la base de données.

Finalement, une autre option intéressante, bien que non prioritaire, serait la possibilité de déterminer tous les résidus d'ARN et de protéines avoisinant un résidu ou un ensemble de résidus donnés dans un rayon de distance donné. Un tel ajout serait grandement utile pour étudier l'interaction d'un motif donné avec son entourage, entre autres dans le contexte des interactions ARN-protéines.

Conclusion

Nous avons implémenté une base de données de motifs qui ont fait l'objet d'étude au LBIT. Cette base de données devrait croître à plus ou moins long terme puisque nous étudions constamment de nouveaux motifs structuraux et que la base de données PDB s'accroît perpétuellement. *MC-Map* est un portail sur cette base de données. Les structures PDB et motifs structuraux y sont stockés dans un format de base de données, ce qui permet de plus facilement manipuler l'information de structure, en plus d'offrir une flexibilité à intégrer de nouveaux types de données. *MC-Map* associe des motifs à des structures PDB et affiche l'information sous forme de cartes d'ARN. Ces cartes d'ARN sont des représentations 2-D de brins d'ARN où chaque instance de motif est marquée par une barre colorée. Les cartes d'ARN permettent de situer des motifs dans un plan 2-D et donnent une meilleure compréhension sur la fonction de ces motifs. Nous souhaiterions dans l'avenir pouvoir intégrer d'autres sources d'information, soit en ajoutant des données de structure supplémentaires à la base de données, tel de l'information d'annotation, soit en reliant *MC-Map* à d'autres serveur Web de structure d'ARN. Nous souhaitons également créer une version de *MC-Map* qui soit téléchargeable et qui puisse fonctionner de manière locale, afin que tout utilisateur puisse étudier ses propres motifs sans être connecté à un réseau.

Bibliographie

1 Nagaswamy U., Voss N., Zhang Z. et Fox G. E., *Database of non-canonical base pairs found in known RNA structures,* NAR, 1999

2 Leontis, N.B., Westhof, E., *Geometric nomenclature and classification of RNA base pairs*, RNA, 2001

3 Hanlon, S., *The importance of london dispersion forces in the maintenance of deoxyribonucleic acid double helix*, Biochem. Biophysic. Res. Commun. 23, 861-867, 1966

4 Sarai, A., Mazur, J., Nussinov, R. et Jermigan, R.L., *Origin of DNA helical structure and its sequence dependence*, Biochemistry, 27, 8498-8502, 1988

5 Berman, H.M., Westbrook, J., Feng, Z., Gilliland, G., Bhat, T.N., Weissig, H., Shindyalov, I.N. et Bourne, P.E., *The Protein Data Bank*, Nucleic Acids Res. 28. 235-242, 2000

6 Saenger, W., *Principles of Nucleic Acid Structure*, Springer-Verlag, New York, USA, 1984

7 Ban, N., Nissen, P., Hansen, J., Moore, P.B. et Steitz, T.A., *The complete atomic structure of the large ribosomal subunit at 2.4A resolution*, Science, 289, 905-920 (2000)

8 Wimberly, B.T., Brodersen, D.E., Clemons, W.M.Jr, Morgan-Warren, R.J., Carter, A.P., Vonrhein, C., Hartsch, T. and Ramakrishnan, V., *Structure of the 30S ribosomal subunit*, Nature, 407, 327–339, (2000)

9 Carter, A.P., Celmons,W.M., Brodersen,D.E., Morgan-Warren,R.J., Wimberly,B.T. and Ramakrishnam,V., *Functional insights from the structure of the 30S ribosomal subunit and its interactions with antibiotics*, Nature, 407, 340–348, (2000)

10 Gendron P., Lemieux S. et Major F., *Quantitative analysis of nucleic acid three-dimensional structures*, Journal of Molecular Biology, 308(5): 919-936, 2001

11 Larose M., Gendron P. et Major F., *MC-Search: a three dimensional RNA pattern matching tool*, RNA 2005 : Tenth annual meeting of the RNA SOCIETY, May 24-29, 2005

12 Ullmann J.R., *An Algorithm for Subgraph Isomorphism*, Journal of the Association for Computing Machinery, 23, 31-42, 1976

13 Klosterman PS, Tamura M, Holbrook SR, Brenner SE., *SCOR: a structural classification of RNA database*, Nucleic Acids Res. 30. 392-394, 2002

Annexe A : Architecture de la base de données

```
--
-- Host: mysql    Database: lbit_mcmap
---------------------------------------------------------
--
DROP DATABASE IF EXISTS lbit_mcmap;
CREATE DATABASE lbit_mcmap;
USE lbit_mcmap;


CREATE TABLE Atom (
  AtomId BIGINT signed NOT NULL auto_increment,
  AtomName varchar(4) NOT NULL default '',
  ResId BIGINT signed NOT NULL,
  Xvalue varchar(8) NOT NULL,
  Yvalue varchar(8) NOT NULL,
  Zvalue varchar(8) NOT NULL,
  Ovalue varchar(8) NOT NULL,
  Bvalue varchar(8) NOT NULL,
  PRIMARY KEY  (AtomId),
  KEY ResId (ResId)
) TYPE=MyISAM;


CREATE TABLE Chain (
  ChainId BIGINT signed NOT NULL auto_increment,
  MolId int(3) signed NOT NULL,
  Name varchar(3),
  PdbName varchar(4) NOT NULL default '',
  Type varchar(4) NOT NULL default '',
  Description varchar(128),
  ScienceName varchar(64),
  CommonName varchar(32),
  PRIMARY KEY (ChainId)
) TYPE=MyISAM;


CREATE TABLE Dataset (
  DatasetId int(8) signed NOT NULL auto_increment,
  Name varchar(32),
  PRIMARY KEY (DatasetId),
  KEY Name (Name)
) TYPE=MyISAM;


CREATE TABLE DatasetPdb (
  DatasetId int(8),
  PdbName varchar(4),
  PRIMARY KEY (DatasetId, PdbName)
) TYPE=MyISAM;
```

```
CREATE TABLE Motif (
  MotifId BIGINT signed NOT NULL auto_increment,
  Name varchar(32),
  Comment varchar(255),
  Date DATETIME,
  TextData TEXT,
  PRIMARY KEY (MotifId)
) TYPE=MyISAM;


CREATE TABLE MotifInst (
  MotifInstId BIGINT signed NOT NULL auto_increment,
  MotifId BIGINT signed,
  PdbName varchar(4),
  MoleculeId int(8),
  InstanceId int(8),
  PictureData longblob,
  ThumbData blob,
  PRIMARY KEY (MotifInstId)
) TYPE=MyISAM;


CREATE TABLE MotifPdb (
  MotifId BIGINT signed,
  PdbName varchar(4),
  Total int(8) default '0',
  Status int(1) signed default '0',
  Date DATETIME,
  PRIMARY KEY (MotifId, PdbName)
) TYPE=MyISAM;


CREATE TABLE MotifRes (
  MotifInstId BIGINT signed,
  ResId BIGINT signed,
  PRIMARY KEY (MotifInstId, ResId)
) TYPE=MyISAM;


CREATE TABLE Motifset (
  MotifsetId int(8) signed NOT NULL auto_increment,
  Name varchar(32),
  PRIMARY KEY (MotifsetId),
  KEY Name (Name)
) TYPE=MyISAM;


CREATE TABLE MotifsetMotif (
  MotifsetId int(8),
  MotifId BIGINT signed,
  PRIMARY KEY (MotifsetId, MotifId)
) TYPE=MyISAM;
```

```
CREATE TABLE Pdb (
  PdbName varchar(4) NOT NULL default '',
  Header varchar(64) NOT NULL default '',
  Title varchar(250) NOT NULL default '',
  Date DATE,
  Experience varchar(64) NOT NULL default '',
  Resolution decimal(4,2) default '0.0',
  PRIMARY KEY (PdbName)
) TYPE=MyISAM;


CREATE TABLE Project (
  ProjectId BIGINT signed NOT NULL auto_increment,
  Name varchar(32),
  Comment varchar(128),
  Date DATETIME,
  PRIMARY KEY (ProjectId)
) TYPE=MyISAM;


CREATE TABLE ProjectMotif (
  ProjectId BIGINT signed,
  MotifId BIGINT signed,
  MotifsetId BIGINT signed,
  Color varchar(6),
  PRIMARY KEY (ProjectId, MotifId, MotifsetId)
) TYPE=MyISAM;


CREATE TABLE ProjectPdb (
  ProjectId BIGINT signed,
  PdbName varchar(4),
  DatasetId int(8),
  PRIMARY KEY (ProjectId, PdbName, DatasetId)
) TYPE=MyISAM;


CREATE TABLE Residue (
  ResId BIGINT signed NOT NULL auto_increment,
  ResNum varchar(8) NOT NULL default '',
  ResName varchar(4) NOT NULL default '',
  ChainId BIGINT signed,
  PRIMARY KEY (ResId)
) TYPE=MyISAM;
```

Annexe B : Utilisation du script d'intégration *pdb2db*

Voila ci-dessous quelques exemples d'utilisation du script d'intégration *pdb2db*. Il est important de noter qu'afin d'utiliser le script d'intégration, il faut en un premier temps que celui-ci fasse partie du chemin d'exécution ('path') et avoir un accès à la base de données de *MC-Map*. L'application *pdb2db* est présentement uniquement disponible sur l'espace Web du LBIT, plus précisément dans le sous-répertoire 'script' du répertoire racine de *MC-Map*.

`pdb2db 1JJ2.pdb`	Intègre le fichier de structure PDB 1JJ2 dans la base de données MySQL. On assume ici que le fichier PDB existe dans le répertoire courant.
`pdb2db -p 1JJ2`	L'option p vient renseigner que le fichier de structure 1JJ2 n'existe pas sur le répertoire courant et que par conséquent, il doit être téléchargé du site Web PDB afin de l'intégrer à la base de données.
`pdb2db -p -m 1JJ2`	L'option m déclenche une recherche sur la structure PDB (qui ici est également à télécharger du site Web PDB), sur tous les motifs existants de la base de données.
`pdb2db 1JJ2_1-02.pdb -d descripteur.mcc`	Intègre l'instance de motif '1JJ2_1-02.pdb' générée par *MC-Search*, correspondant au motif décrit par le fichier descripteur 'descripteur.mcc'. Le fichier descripteur doit respecter un certain format afin que le motif décrit puisse être reconnu par l'application. L'interpréteur vérifie également si le format du nom de fichier est exact et dans ce cas-ci, assigne à l'instance #2 de la première molécule

	du fichier de structure 1JJ2, les résidus correspondant. Il est par après impossible d'intégrer la même instance de motif.
`pdb2db 1JJ2_1-02.gif -d descripteur.mcc`	Intègre l'image d'aperçu correspondant à l'instance de motif '1JJ2_1-02.pdb' générée par *MC-Search* correspondant au motif décrit par le fichier descripteur 'descripteur.mcc'. Le fichier descripteur doit respecter un certain format afin que le motif décrit puisse être reconnu par l'application. Ne fonctionne que si l'instance PDB a déjà été intégrée auparavant (voir commande précédente). Une image d'aperçu peut être mise à jour contrairement à une instance de motif PDB.
`pdb2db 1JJ2_1-02.jpg -d descripteur.mcc`	Intègre le graphe d'annotation correspondant à l'instance de motif '1JJ2_1-02.pdb' générée par *MC-Search*. Identique à la commande précédente.
`pdb2db -d 707`	Extrait de la base de données le fichier descripteur du motif dont l'identificateur est 707. Le fichier 'desc000707.mcc' sera alors généré dans le répertoire courant.
`pdb2db -s -d 707`	Recherche le motif dont l'identificateur est 707 dans toutes les structures PDB. S'applique seulement aux recherches de motif qui n'ont jamais été faites et qui ne sont pas en cours d'exécution.
`pdb2db -s`	Recherche tous les motifs de la base de données dans toutes les structures PDB de la base de données.

	S'applique seulement aux recherches de motif qui n'ont jamais été faites et qui ne sont pas en cours d'exécution. L'usager doit par après confirmer la commande car cette recherche peut être particulièrement longue.